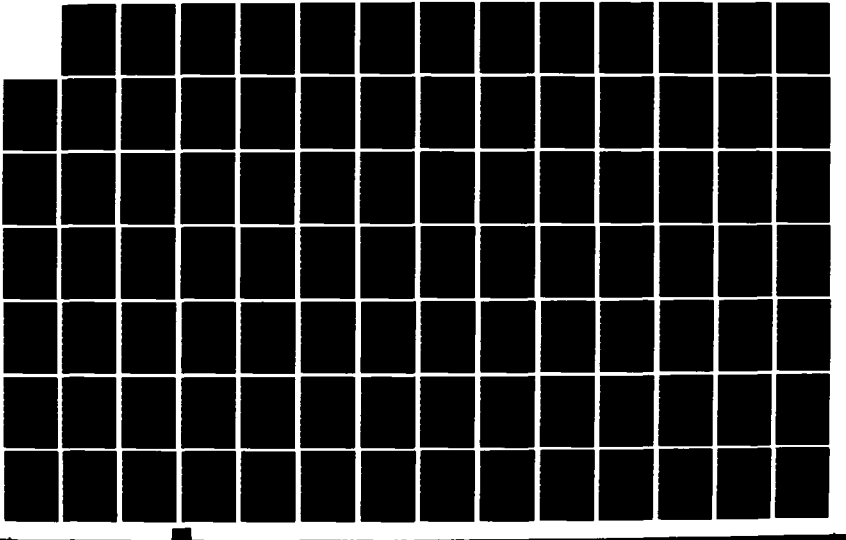
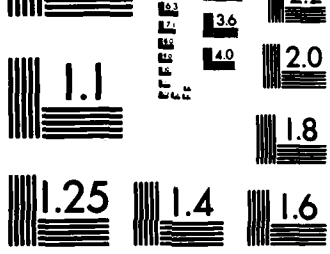
AD-A131 282 SUBMERGED SHOCK RESPONSE OF A LINEARLY ELASTIC SHELL OF REVOLUTION CONTRI..(U) WEIDLINGER ASSOCIATES NEW YORK R YASUDEVAN ET AL. 01 MAY 82 DNA-TR-81-184 1/2

UNCLASSIFIED DNA001-81-C-0048 F/G 20/11 NL





(12)

DNA-TR-81-184

SUBMERGED SHOCK RESPONSE OF A LINEARLY ELASTIC SHELL OF REVOLUTION CONTAINING INTERNAL STRUCTURE

User's Manual for the ELSHOK Code

R. Vasudevan
D. Ranlet
Weidlinger Associates
333 Seventh Avenue
New York, New York 10001

1 May 1982

Technical Report

CONTRACT No. DNA 001-81-C-0048

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK WAS SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B344081466 Y99QAXSF00002 H2590D.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, DC 20305

Destroy this report when it is no longer needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY, ATTN: STTI, WASHINGTON, D.C. 20305, IF YOUR ADDRESS IS INCORRECT, IF YOU WISH TO BE DELETED FROM THE DISTRIBUTION LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA-TR-81-184	2. GOVT ACCESSION NO. AD-A131 282	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SUBMERGED SHOCK RESPONSE OF A LINEARLY ELASTIC SHELL OF REVOLUTION CONTAINING INTERNAL STRUCTURE - User's Manual for the ELSHOK Code.		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R. Vasudevan and D. Ranlet		8. CONTRACT OR GRANT NUMBER(s) DNA 001-81-C-0048
9. PERFORMING ORGANIZATION NAME AND ADDRESS Weidlinger Associates 333 Seventh Avenue New York, New York 10001		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task Y99QAXSF-00002
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D.C. 20305		12. REPORT DATE 1 May 1982 13. NUMBER OF PAGES 144
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This work was sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B344081466 Y99QAXSF00002 H2590D.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Structure-Fluid Interaction
Underwater Shock
Elastic Shock Analysis
Dynamic Response
Substructuring
Transient Response

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

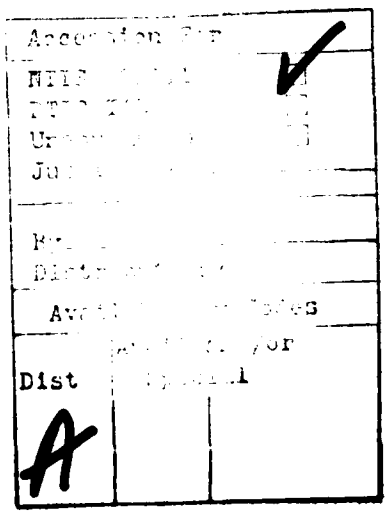
Presented herein is a user's manual for the ELSHOK computer code. This computer program calculates the transient response of a submerged ring-stiffened shell of revolution, with or without internal structure, to an underwater shock wave having an arbitrary direction of impingement. Linearly elastic structures are considered, and the surrounding fluid is treated as an infinite acoustic medium. Modal analysis is employed for the structure, while the analysis of the fluid is based upon the Doubly Asymptotic Approximation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data F

SUMMARY

Presented herein is a user's manual for the ELSHOK computer code. This computer program calculates the transient response of a submerged ring-stiffened shell of revolution, with or without internal structure, to an underwater shock wave having an arbitrary direction of impingement. Linearly elastic structures are considered, and the surrounding fluid is treated as an infinite acoustic medium. Modal structural analysis is employed throughout. When the shell contains internal structure, a substructuring technique is applied in which the modes of the empty ring-stiffened shell and the fixed-base modes of each piece of internal structure are coupled through the use of dynamic boundary conditions. The structure-fluid interaction is approximated by means of the Doubly Asymptotic Approximation expressed in terms of functions which are orthogonal over the wet surface of the submerged shell.



Conversion factors for U.S. customary to metric (SI) units of measurement

To Convert From	To	Multiply By
angstrom	meters (m)	1.000 000 X E -10
atmosphere (normal)	kilo pascal (kPa)	1.013 25 X E +2
bar	kilo pascal (kPa)	1.000 000 X E +2
barn	meter^2 (m^2)	1.000 000 X E -28
British thermal unit (thermochemical)	joule (J)	1.054 350 X E +3
calorie (thermochemical)	joule (J)	4.184 000
cal (thermochemical)/cm^2	mega joule/m^2 (MJ/m^2)	4.184 000 X E -2
curie	*giga becquerel (GBq)	3.700 000 X E +1
degree (angle)	radian (rad)	1.745 329 X E -2
degree Fahrenheit	degree kelvin (K)	$t_K = (t°f + 459.67)/1.8$
electron volt	joule (J)	1.602 19 X E -19
erg	joule (J)	1.000 000 X E -7
erg/second	watt (W)	1.000 000 X E -7
foot	meter (m)	3.048 000 X E -1
foot-pound-force	joule (J)	1.355 818
gallon (U.S. liquid)	meter^3 (m^3)	3.785 412 X E -3
inch	meter (m)	2.540 000 X E -2
jerk	joule (J)	1.000 000 X E +9
joule/kilogram (J/kg) (radiation dose absorbed)	Gray (Gy)	1.000 000
kilotons	terajoules	4.183
kip (1000 lbf)	newton (N)	4.448 222 X E +3
kip/inch^2 (ksi)	kilo pascal (kPa)	6.894 757 X E +3
ktap	newton-second/m^2 (N-s/m^2)	1.000 000 X E +2
micron	meter (m)	1.000 000 X E -6
mil	meter (m)	2.540 000 X E -5
mile (international)	meter (m)	1.609 344 X E +3
ounce	kilogram (kg)	2.834 952 X E -2
pound-force (lbs avoirdupois)	newton (N)	4.448 222
pound-force inch	newton-meter (N·m)	1.129 848 X E -1
pound-force/inch	newton/meter (N/m)	1.751 268 X E +2
pound-force/foot^2	kilo pascal (kPa)	4.788 026 X E -2
pound-force/inch^2 (psi)	kilo pascal (kPa)	6.894 757
pound-mass (lbm avoirdupois)	kilogram (kg)	4.535 924 X E -1
pound-mass-foot^2 (moment of inertia)	kilogram-meter^2 (kg·m^2)	4.214 011 X E -2
pound-mass/foot^3	kilogram/meter^3 (kg/m^3)	1.601 846 X E +1
rad (radiation dose absorbed)	**Gray (Gy)	1.000 000 X E -2
roentgen	coulomb/kilogram (C/kg)	2.579 760 X E -4
shake	second (s)	1.000 000 X E -8
slug	kilogram (kg)	1.459 390 X E +1
torr (mm Hg, 0° C)	kilo pascal (kPa)	1.333 22 X E -1

*The becquerel (Bq) is the SI unit of radioactivity; 1 Bq = 1 event/s.
**The Gray (Gy) is the SI unit of absorbed radiation.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

I. INTRODUCTION

The development of analytical methods and computer codes for the analysis of the response of submerged structures to underwater shock waves has made considerable progress in recent years. In particular, Weidlinger Associates, under the sponsorship of the Defense Nuclear Agency (DNA) and the Office of Naval Research (ONR), has developed the ELSHOK (ELASTIC SHOCK) computer code for analyzing such problems when only elastic structural deformations are involved. For the case of more intense loadings in which elasto-plastic structural deformations occur, the EPSA code of Ref. [1] may be used.

The present report describes the ELSHOK computer code. This computer program calculates the transient response of a submerged ring-stiffened shell of revolution, with or without internal structure, to an underwater shock wave emanating from an explosive source placed at an arbitrary location away from the shell. Linearly elastic structures are considered, and the surrounding fluid is treated as an infinite acoustic medium. Modal structural analysis is employed throughout. When the shell contains internal structure, the substructuring technique of Refs. [2] and [3] is applied in which the free-free modes of the empty ring-stiffened shell and the fixed-base modes of each piece of internal structure are coupled through the use of dynamic boundary conditions. The structure-fluid interaction is approximated in ELSHOK by means of the Doubly Asymptotic Approximation (DAA) of Ref. [4], expressed in terms of functions which are orthogonal over the wet surface of the submerged shell (Ref. [2]).

The development of ELSHOK has been in tandem with an experimental program aimed at providing a high quality data base for judging computerized prediction methods. This combined analytical/experimental

effort continues to address the submerged shock response of structures of increasing complexity. To date, comparisons of test data and calculations performed using ELSHOK have shown that ELSHOK is a viable design/analysis tool.

Documentation for the current operational version of ELSHOK is provided in this manual for theory, code design and implementation, input data, operating instructions, data management, and two sample problems. It is hoped that the information supplied here is sufficiently detailed to enable a new user to perform an ELSHOK calculation. The authors wish to broaden the ELSHOK user community and to establish communication links between users and developers so that the ELSHOK code may be enhanced to satisfy future requirements.

It should be pointed out at this time that modifications to ELSHOK have been made to permit the use of geometrically non-linear members as shock mounts. These mounts may serve as connections between (1) internal structure and the ring-stiffened shell and (2) pieces of internal structure. Thus, the next generation of the ELSHOK code will allow hard-mounted, soft (spring)-mounted, and shock-mounted internal appendages to be treated in a routine manner. The improved version of ELSHOK will be released after the completion of extensive checking, and a revised user's manual will then be issued.

II. THEORY

a. General Information

The ELSHOK computer code is a suite of computer programs assembled and developed to calculate the submerged shock response of a linearly elastic structure subjected to an underwater shock wave. Specifically, a ring-stiffened shell of revolution of finite length, with or without internal structure, is considered to be immersed, initially at rest, in an infinite acoustic fluid and to be excited by an acoustic spherical wave, of arbitrary pressure profile, emanating from an explosive source located at an arbitrary point away from the structure. It is required to compute the transient response of the shell and any internal appendages and to provide results in the form of printed and/or plotted velocity-time histories and associated pseudo-velocity shock spectra.

In ELSHOK, the structure-fluid interaction is approximated by means of the DAA of Ref. [4]. Following Ref. [2], the form of the DAA used is that obtained when the normal fluid displacement of the structure-fluid interface is expanded in a series of functions (surface expansion functions) which are orthogonal over the wet surface of the structure. The elements of the matrices in the resulting DAA are obtained by matching exact pressure-velocity relations at zero and infinite frequencies. Thus, in transient problems, the DAA yields exact results at early and late times and, by the nature of its formulation, brings about a smooth transition between these two limits. The DAA accounts for the effects of the fluid by quantities defined solely on the wet surface of the submerged body and may be thought of as uncoupling the fluid field from the structural field.

The ELSHOK code determines the transient response of the submerged structural system described above by using component modal analysis, or substructuring, as discussed in Ref. [3]. In this technique, the complete structural system is assumed to be comprised of a main structure S (the shell) and any attached substructures or internal appendages σ. The vibration modes of each component are determined separately, and the equations of motion of the system S + σ are obtained by enforcing compatibility of deformation at the points of attachment. The implementation of substructuring in ELSHOK also allows for the case of no attached appendages (i.e., for an empty shell S).

b. Response Equations

As shown in Refs. [2] and [3], the use of the DAA permits the equations governing the shock response of a submerged body S, with internal structure σ, to be written as

$$\mu_j \ddot{q}_j + \mu_j \omega_j^2 q_j = \sum_{i=1}^{\nu_F} \lambda_{ij} (\hat{Q}_i + \hat{Q}_{Ii}) + Q_{\sigma j} \qquad (j = 1,2,\ldots,\nu_S) \qquad (1)$$

$$\dot{q}_j(0) = q_j(0) = 0 \qquad (j = 1,2,\ldots,\nu_S) \qquad (2)$$

$$\hat{Q}_j = \rho c \hat{\mu}_j \left[\hat{U}_{Ij} + \sum_{i=1}^{\nu_F} \hat{e}_{ji} \int_0^t \hat{Q}_i dt - \sum_{i=1}^{\nu_S} \lambda_{ji} \dot{q}_i\right] \qquad (j = 1,2,\ldots,\nu_F) \qquad (3)$$

$$\left[\int_0^t \hat{Q}_j dt\right]_{t=0} = 0 \qquad (j = 1,2,\ldots,\nu_F) \qquad (4)$$

where the q_j are the generalized coordinates of the main structure S, ω_j and μ_j are, respectively, the free-free natural frequencies and corresponding generalized masses of S, ρ and c denote, respectively, the mass density of and speed of sound in the fluid, ν_S and ν_F represent, respectively, the number of shell modes and surface expansion functions used, and a dot denotes differentiation with respect to time t. Additional symbols appearing in the above equations will be defined below. It may be observed that the effect of the fluid in the interaction problem, as described by the DAA of Eq. (3), is simply to apply an additional loading on the shell S, resulting from the $\hat{Q}_i$ terms in Eq. (1). No further analysis of the fluid field is required during the structure-fluid interaction problem.

The elements of the fluid-shell transformation matrix appearing in Eqs. (1) and (3) are given by

$$\lambda_{ji} = \frac{1}{\hat{\mu}_j} \int_A \phi_i^n \psi_j \, dA \qquad (i = 1,2,\ldots,\nu_S \; ; \; j = 1,2,\ldots,\nu_F) \tag{5}$$

where ϕ_i^n are the normal components of the free-free in-vacuo mode shapes of the empty structure S (i.e., without internal structure), ψ_j are the surface expansion functions, A is the wet surface area of the body S, and the $\hat{\mu}_j$ are normalization constants defined by

$$\hat{\mu}_i \delta_{ij} = \int_A \psi_i \psi_j \, dA = A\delta_{ij} \qquad (i,j = 1,2,\ldots,\nu_F) \tag{6}$$

in which δ_{ij} is the Kronecker delta.

If p_I and u_I represent, respectively, the incident pressure and normal fluid particle velocity on the wet surface of S, the expansion coefficients associated with the incident shock wave [Eqs. (1) and (3)] may be expressed as

$$\hat{Q}_{Ij} = \int_A p_I \psi_j \, dA \qquad (j = 1,2,\ldots,\nu_F) \tag{7}$$

$$\hat{U}_{Ij} = \frac{1}{\hat{\mu}_j} \int_A u_I \psi_j \, dA \qquad (j = 1,2,\ldots,\nu_F) \tag{8}$$

The $\hat{e}_{ji}$ terms in Eq. (3) are the elements of the inverse of the symmetrical virtual mass matrix formulated in terms of the ψ_j. Moreover, the $Q_{\sigma j}$ in Eq. (1) are the generalized forces exerted on the body S by the internal structure σ. These terms contain the effects of the elastic reactions of c on S and of the inertial forces due to any concentrated masses joined to S. General matrix equations for the $Q_{\sigma j}$ are given in Ref. [3].

If a substructure σ is attached to S without the use of intervening energy-absorbing devices or shock mounts, Ref. [3] shows that the physical degrees of freedom of σ may be partitioned into the two sets

$$\bar{X}_{\sigma j} = \sum_{i=1}^{\nu_S} C_{ji} q_i \qquad (j = 1,2,\ldots,\bar{N}) \tag{9}$$

$$\hat{X}_{\sigma j} = \sum_{i=1}^{\nu_\sigma} \hat{\phi}_{\sigma ji} q_{\sigma i} + \sum_{i=1}^{\bar{N}} \hat{g}_{ji} \bar{X}_{\sigma i} \qquad (j = 1,2,\ldots,\hat{N}) \tag{10}$$

where the $\bar{X}_{\sigma j}$ are the $\bar{N}$ physical degrees of freedom of σ which are constrained to move with S, the $\hat{X}_{\sigma j}$ are the remaining $\hat{N}$ active physical degrees of freedom of σ, and the $q_{\sigma i}$ are the ν_σ generalized coordinates of σ. In Eqs. (9) and (10), the C_{ji} are coefficients obtained from the modes of S resulting from the enforcement of compatibility of deformation at the points of attachment of σ to S, the $\hat{\phi}_{\sigma ji}$ are the elements of the fixed-base modes of σ, and the $\hat{g}_{ji}$ are influence coefficients (sometimes termed "constraint modes") accounting for the support or base motions of σ.

As discussed in Ref. [3], the partitioning of Eqs. (9) and (10) leads to the following equations for the generalized coordinates of an elastic internal appendage σ connected to a main structure S:

$$\ddot{q}_{\sigma j} + \omega_{\sigma j}^2 q_{\sigma j} = - \sum_{k=1}^{\bar{N}} G_{jk} \sum_{i=1}^{\nu_S} C_{ki} \ddot{q}_i \qquad (j = 1,2,\ldots,\nu_F) \tag{11}$$

$$\dot{q}_{\sigma j}(0) = q_{\sigma j}(0) = 0 \qquad (j = 1,2,\ldots,\nu_F) \tag{12}$$

in which the $\omega_{\sigma j}$ are the fixed-base natural frequencies of the internal structure σ and the G_{jk} are terms resulting from the support on base motions of σ. Reference [3] contains general matrix equations for the G_{jk}, the C_{jk}, and the $\hat{g}_{jk}$. Response equations corresponding to Eqs. (11) and (12) are required for each substructure included in a given analysis.

c. Surface Expansion Functions

Surface expansion functions are spatial functions having the property of orthogonality over the wet surface of a submerged body. As pointed out in Ref. [2], when surface expansion functions are used in a series expansion of the normal displacement of the wet surface, they lead to matrices in the DAA which are well-conditioned for inversion. The surface expansion functions used in any calculation must be capable of representing the normal displacement of the structure-fluid interface.

In ELSHOK, the determination of the surface expansion functions requires the wet surface of the structure under study to be partitioned into a central region (which need not be a right circular cylinder) and two end closures of arbitrary revolved shape, as shown in Fig. 1. When symmetry is employed along the length of the structure, only the left end closure must be specified. Since ELSHOK allows a user to highlight a portion of the structure when localized responses are expected, a special section of the structure, called the "compartment region" or "compartment model", may also be specified. The use of the compartment region is discussed in detail in Appendices A and B.

Points on the wet surface are located by specifying the meridional arc length s, measured along the reference curve used to generate the wet surface of revolution, and the circumferential coordinate angle θ (Fig. 1). Following the shell-of-revolution analysis employed by ELSHOK for the main body S, the θ-dependence of the surface expansion functions is eliminated by means of separation of variables. Thus, the circumferential distributions of these functions are taken to be $\cos N\theta$ or $\sin N\theta$, as appropriate, where N is the integral number of circumferential waves or

circumferential harmonics. The meridional distributions of the surface expansion functions may then be obtained for one given value of N at a time.

For N=0 (axisymmetrical) and $N \geq 1$, the meridional distribution of a convenient set of functions for generating orthogonal functions capable of describing the normal motion of the central region of the structure illustrated in Fig. 1 may be written symbolically as

$$\xi_j = \cos (M \pi z/L) \quad (M = 0,1,2,\ldots) \text{ on central region}$$
$$\xi_j = 0 \text{ elsewhere} \tag{13}$$

where L is the meridional arc length of the central region, M is the number of meridional half-waves in L, and z is the meridional coordinate measured from the left end of the central region ($0 \leq z \leq L$).

For N=0, the meridional distribution of a convenient set of functions for generating orthogonal functions capable of describing the normal motion of the end closures may be written symbolically as

$$\xi_k = a_c \cos \left(\frac{\bar{M}\pi\bar{s}}{2\ell}\right) \quad (\bar{M} = 0,2,4,\ldots) \text{ on each end closure}$$
$$\xi_k = 0 \text{ elsewhere} \tag{14}$$

in which ℓ is the meridional arc length of the generator of an end closure, $\bar{M}$ is the number of quarter-waves in ℓ, $a_c = \pm 1$ is a factor used to construct symmetrical and antisymmetrical functions, and $\bar{s}$ is the meridional coordinate measured from the appropriate pole ($0 \leq \bar{s} \leq \ell$). For $N \geq 1$, Eq. (14) must be replaced by

$$\xi_k = a_c \sin\left(\frac{\bar{M}\pi\bar{s}}{2\ell}\right) \quad (\bar{M} = 1,3,5,\ldots) \text{ on each end closure}$$

$$\xi_k = 0 \text{ elsewhere} \tag{15}$$

in order to avoid a directional singularity at the pole of each end closure when the circumferential distribution is supplied.

For N=0 (axisymmetrical) and N ≥ 1, the meridional distribution of a convenient set of functions for generating orthogonal functions capable of describing the localized normal motion of the compartment region may be written symbolically as

$$\xi_j = \sin\left(M\pi\,\bar{z}/\bar{L}\right) \quad (M = 1,2,3,\ldots) \text{ on compartment region}$$

$$\xi_j = 0 \text{ elsewhere} \tag{16}$$

where $\bar{L}$ is the meridional arc length of the compartment region, M is the number of meridional half-waves in $\bar{L}$, and $\bar{z}$ is the meridional coordinate measured from the left end of the compartment region ($0 \leq \bar{z} \leq \bar{L}$).

In the usual application of ELSHOK, a central region and two end closures (or one if symmetry is employed) are considered without the use of a compartment region. For this case, the surface expansion functions are generated from the convenient functions of Eqs. (13)-(15) for each required value of N. When a compartment region is employed in addition to the central and end-closure regions, the representation of the normal motion of the wet surface of the submerged structure changes. For N=0,2,3,..., the surface expansion functions are generated from Eq. (16) and are employed to capture local responses. For N=1, the surface expansion functions are generated from Eqs. (13)-(16) and are employed to capture both global and local responses.

Once the decision has been made about which of the above sets of convenient functions are to be used to describe the normal displacement of the structure-fluid interface in a given problem, the surface expansion functions may then be generated. The ELSHOK code accomplishes this task by means of the Gram-Schmidt orthogonalization process (Ref. [5]). Normalization of the surface expansion functions is done according to Eq.(6).

d. Incident Loading

As may be seen from Eqs. (1) - (8), the solution of the governing equations requires a knowledge of both the incident pressure and the incident fluid particle velocity applied by the shock wave to the wet surface of the submerged structure. In ELSHOK, the incident wave is assumed to be a spherical wave propagating through a linear acoustic fluid. Thus, the fluid particle velocity may be determined from a knowledge of the incident pressure. Two representations of the incident pressure are included in ELSHOK : a known pressure profile such as that determined by experiment and the empirical decaying exponential form of Ref. [6] for spherical charges. Neither loading option considers the effects of cavitation.

For incident loading specified in terms of a known pressure profile, the incident pressure and normal fluid particle velocity at a given point on the wet surface may be written as

$$p_I(R,t) = \frac{S_o}{R} \; P(t - \frac{R-S_o}{c}) \; H(t - \frac{R-S_o}{c}) \tag{17}$$

$$u_I(R,t) = [\frac{1}{\rho c} p_I(R,t) + \frac{1}{\rho R} \int_o^t p_I(R,\tau) \, d\tau] \cos \phi \tag{18}$$

where R is the distance from the origin of the incident spherical wave to the point of interest, S_o is the stand-off distance (the distance between the origin of the incident wave and the closest point on the body), H is the Heaviside unit function, P is the known pressure-time function at the point where the shock wave first contacts the submerged body ($R = S_o$), and ϕ is the angle between the spherical wave front and the inward normal to the wet surface of the submerged body at the point of interest. It may be observed that the time t is measured from the first arrival of the shock front at the surface of the submerged structure.

When the decaying exponential form is employed, the incident pressure is given by

$$p_I(R,t) = P_o(R,W) \exp\left[-\left(t - \frac{R-S_o}{c}\right) / \theta_o(R,W)\right] H\left(t - \frac{R-S_o}{c}\right) \qquad (19)$$

in which $P_o(R,W)$ is the initial peak pressure, $\theta_o(R,W)$ is the time constant of exponential decay, and W is the weight of the spherical charge used to produce the shock wave. For this loading option, the incident normal fluid particle velocity is approximated by using Eq. (19) in Eq. (18), yielding

$$u_I(R,t) = \left[\frac{p_I}{\rho c} + \frac{\theta_o}{\rho R}(P_o - p_I)\right] H\left(t - \frac{R-S_o}{c}\right) \cos\phi \qquad (20)$$

Such a procedure for determining the incident fluid particle velocity is approximate because, as shown in Ref. [6], the spatial decay of P_o differs slightly from the 1/R decay in a true spherical wave.

III. ORGANIZATION AND OPERATION

a. Outline

The ELSHOK computer code may be used to calculate the linearly elastic shock response of a submerged ring-stiffened shell of revolution containing internal structure. A typical shell-substructure configuration is displayed in Fig. 1. The analysis of such a problem using ELSHOK is based upon a modal substructuring procedure which does not involve the modes and natural frequencies of the combined structural system (shell and appendages). In addition, as discussed in Ref. [3], a system stiffness matrix is not required, since the substructuring technique employs interaction forces and moments and compatibility of deformation at the shell-substructure junctions to solve the dynamic response problem. A modal mass matrix of size ν_S by ν_S is required for those problems involving concentrated or lumped masses attached to the shell.

Separate modal analyses of the shell S and any internal structure σ are required when using ELSHOK. The shell, a body of revolution, is analyzed using the BOSOR4 finite difference code of Ref. [7]. The internal structure, if any, is analyzed by means of the finite element code SAPIV of Ref. [8]. User familiarity with these codes is assumed. However, since some modifications to BOSOR4 and SAPIV have been made to allow the use of these codes in the ELSHOK program suite, some information about them will be given as needed.

The ELSHOK code is written in FORTRAN IV for use on Control Data Corporation (CDC) computers. The code is modular in construction and operates on CDC SCOPE, KRONOS, NOS, and NOS/BE operating systems of various

levels. Machine dependency is minimal and the transfer of ELSHOK to computers of other makes is not a major problem.

The source decks for the ELSHOK code are maintained in the format of the CDC UPDATE batch editing system (Ref. [9]). Thus, the entire ELSHOK code is stored in a collection of files known as "program libraries". Every card image in a program library is assigned a unique identifier. This enables a user to reference any card image during an UPDATE modification run. Hence, card images may be inserted into or deleted from a program library to alter a program module. The modified program library may then be processed by the FORTRAN compiler and the resulting compiled or object code may be executed. The UPDATE system allows a user great flexibility in the use of ELSHOK.

All of the major components of the ELSHOK code, except BOSOR4, make use of a technique usually referred to as "dynamic allocation of central memory" or "main memory management". In this technique, only those arrays required during a program execution step are stored in a column matrix, called the "master array", allocated to blank common. An individual array is located by specifying the location of its first word within the master array; the location of its last word is determined such that space is allocated for the minimum array size needed for the problem under study. Moreover, all arrays of substantial size are variably or adjustably dimensioned in subroutines using them. Hence, the capacity of a code employing main memory management is limited only by the central memory available to the computer used and not by the number of modes, nodal points, etc, as when arrays are dimensioned to fixed sizes.

The implementation of the dynamic allocation in ELSHOK does not allow the size of the master array in blank common to be changed during program execution. It is, therefore, necessary to dimension the master array to its full size prior to compilation. Thus, two steps are usually required when executing a given component of ELSHOK. First, the needed size of the master array is determined, most often by means of an execution option available with most of the components of ELSHOK. Secondly, the dimension of the master array is set, the program compiled, and the execution begun. Although a certain amount of data transfer to and from a disk file (scratch file) is required for the efficient use of the available computer core via main memory management, the resulting increase in the size of a problem which may be solved by ELSHOK is by far the more important consideration.

The major components of ELSHOK are:

1. BOSOR4 - structural analyzer for shell (Ref. [7])
2. ACESNID - virtual mass processor (Ref. [10])
3. PIFLASH - shell-fluid processor
4. SAPIV - structural analyzer for substructure (Ref. [8])
5. PICRUST - substructure processor
6. USLOB - time integration processor
7. PUSLOB - plotting processor.

Several other minor components will be alluded to at the appropriate times. Each component of the ELSHOK code has been written such that any consistent set of units may be employed throughout the analysis of a given problem. Units may be changed when preparing plots of velocity-time histories and shock spectra.

The use of the above program components in the four phases of an ELSHOK calculation is illustrated in Fig. 2. A brief description of each phase follows.

b. Phase I - Shell and Fluid Data

1. Structural Analysis of Shell

The first step in Phase I is to model the rotationally symmetrical portion of the structure being analyzed (Fig. 1) for use with the BOSOR4 computer code. The resulting idealized mathematical model may contain bulkheads and rings, but must exclude internal appendages and concentrated masses. The BOSOR4 code is quite general and applies to segmented, ring-stiffened, branched shells of revolution having various meridional geometries, wall constructions, and ring reinforcements. Circumferential rings may be treated as discrete stiffeners or represented in the orthotropic approximation (smeared) when closely spaced. Stringers must be treated in the orthotropic approximation to preserve structural axisymmetry. Symmetry about the midpoint of the length may be accounted for by appropriate boundary conditions at the plane of symmetry. In ELSHOK, the mathematical model of the entire main body S (or half, if symmetry is employed) is termed the "full model".

In problems where only a portion of a structure is of interest, as is the case when a localized loading is applied to a section of a structure enclosed by stiff bulkheads, the user may highlight the region of interest. This is accomplished in ELSHOK by employing two different mathematical models of the structure when using BOSOR4: the full model described above and the model of the region of interest, called the

"compartment model". The full model is used to capture the gross effects of the response, such as the rigid body translation and whipping of the entire structure (N = 1) and the rolling and twisting of the entire structure (N = 0 torsional). The compartment model is used to capture the local details of the response in the region of interest (N = 0 breathing, $N \geq 2$). The use of this technique is optional, and the full model alone may be employed for the analysis of the submerged shock response. More information about the use of this approximation may be found in Appendix A.

Once the mathematical representation of the main body S has been decided upon, the in-vacuo free-free modes and natural frequencies of S must be determined using the BOSOR4 code. For compatibility with ELSHOK, a separate BOSOR4 calculation must be performed for each circumferential harmonic N included in the analysis. When only a full model is employed, the breathing modes and any torsional modes required (N = 0) may be determined in one calculation or in separate calculations, as desired. For each BOSOR4 calculation, the resulting shell geometry, material properties, generalized masses, modes, and natural frequencies are written on a file (disk or magnetic tape) called a "shell mode file". Operating instructions for the BOSOR4 code may be found in Appendix A.

2. Calculation of Virtual Mass

The second step in Phase I is the computation of the virtual mass array, which provides the late-time contribution of the DAA. In the ELSHOK suite of programs, this is accomplished by means of the ACESNID (ACCESSION TO INERTIA AND DAMPING) code of Ref. [10]. In ACESNID, the virtual mass array is determined from the solution, based on simple sources, of a

low-frequency steady-state (incompressible) problem in which normal displacements corresponding to surface expansion functions are applied to the surface of a cavity of revolution in the infinite fluid having the same shape and size as the wet surface of the structure under consideration.

Program ACESNID obtains the geometry of the cavity in the fluid from the geometry supplied on a shell mode file for a full model. The surface expansion functions required are computed via the Gram-Schmidt orthogonalization process (Ref. [5]), as discussed earlier. The symmetrical virtual mass array for each circumferential wave number N may be determined in separate calculations, or all of the required values of N may be considered in one calculation, depending on the application. The virtual mass and data needed to re-compute the surface expansion functions when required elsewhere are written on a file, called the "virtual mass file", containing virtual mass data for all values of N used. This file may be prepared by one execution of ACESNID for all values of N considered or by merging the results from separate ACESNID calculations. Appendix B contains operating instructions for the ACESNID code.

3. Completion of Data for Shell and Fluid

The third and final step in Phase I is to complete the data for describing the shell and the fluid. For this purpose, the PIFLASH (PREPARE INPUT FOR FLUID AND SHELL) computer code is provided. This code takes the data from all of the shell mode files and from the virtual mass file and reorganizes it to facilitate the solution of the response equations.

During the execution of PIFLASH, the modes obtained from BOSOR4 are all normalized to the total mass of the entire main body S. Thus, when symmetry

is employed, the mass of the mathematical model is doubled. User supplied input determines how many of the shell modes available are to be provided for the time-history calculation. Any concentrated masses included in the mathematical model of the structure under study are accounted for during the execution of PIFLASH. The PIFLASH code constructs a file, called the "shell-fluid file", containing all of the information needed to describe the shell, any concentrated masses attached to the shell, and the fluid during the subsequent analysis. If no internal appendages or substructures are included in the submerged structure being studied, the shell-fluid file serves as the"input file" for the time-history calculation described later. Detailed information about the use of PIFLASH may be found in Appendix C.

c. Phase II - Substructure Data

1. Structural Analysis of Internal Appendages

The first stage of Phase II consists of the development of a finite element model of each piece of internal structure σ. The SAPIV computer program is included in the ELSHOK suite of computer codes for this purpose. Structural systems comprised of a number of different types of finite elements may be analyzed using SAPIV. The SAPIV library of finite elements contains a three-dimensional truss element, a three-dimensional beam element, a plane stress and plane strain element, a two-dimensional axisymmetric solid, a three-dimensional solid, a thick shell element, a thin plate or thin shell element, a boundary element, and a pipe element. Any concentrated masses attached to the substructures must be accounted for in the mathematical models used with SAPIV. If no substructures are present in a given investigation, Phase II does not apply and must be skipped.

Once the mathematical model of each substructure has been completed, the fixed-base modes and natural frequencies of each substructure must be determined. A separate SAPIV calculation must be performed for each internal appendage. For each such calculation, the resulting geometry, stiffness and mass information, modes, and natural frequencies are written on a file called a "substructure mode file". The manner in which SAPIV must be used for an ELSHOK calculation is described in APPENDIX D.

2. Completion of Data for Substructures

The second and final step in Phase II is to complete the data required for each and every internal appendage. Program PICRUST (PREPARE INPUT FOR CALCULATING RESPONSE OF SUBSTRUCTURE) accomplishes this task. The PICRUST code takes the data from a given substructure mode file, rearranges it, and computes a number of additional items needed to perform the solution of the response equations. A separate calculation is required for each substructure.

For a given substructure, the PICRUST code allows the user to determine how many of the fixed-base modes available are to be provided for the time-history calculation. The user also specifies the connectivity between the shell and the substructure under consideration. During an execution of PICRUST, influence coefficients corresponding to the base or support motions of a substructure (the $\hat{g}_{ji}$ of Eq. (10) or the constraint modes of Ref. [3])are computed. Modal coefficients for the calculation of interaction forces and moments between the shell and a substructure are also determined. For each internal appendage studied, the results of the PICRUST execution are stored on a file called a "substructure file". The substructure files for all substructures in a given

investigation must be merged with the shell-fluid file for use in the subsequent shock response problem, as discussed under Phase III. Operating instructions for the PICRUST code are given in Appendix E.

d. Phase III - Submerged Shock Response

1. Completion of Input File

Almost all of the input data required for the structure-fluid interaction calculation must be supplied in a single file, termed the "input file". For problems without internal appendages, the shell-fluid file described under Phase I is the input file for the shock response calculation. For problems involving substructures, the shell-fluid file must be merged with all of the substructure files of Phase II. This may be easily accomplished by means of simple control card (job control language) operations, or a FORTRAN computer program may be written for this purpose. The usual ELSHOK analysis employs the control card operations, as explained in Appendix F.

2. Time Integration

The ELSHOK suite of computer programs includes the computer code USLOB (UNDERWATER SHOCK LOADING OF BODIES) for solving the governing equations of the current formulation of the structure-fluid interaction problem. The USLOB time integration processor employs the Runge-Kutta technique, in the modification due to Gill (Ref. [11]),to perform the integration in time. In addition to the input file described above, the user must also supply time step information and a description of the shock wave loading. At the discretion of the user, input may be supplied to cause the USLOB code to provide printed velocity-time histories for points on the shell and any internal appendages.

To permit the production of plotted velocity-time histories in Phase IV of an ELSHOK calculation, program USLOB provides up to two post-processing files. The first file, called the "generalized velocity file", contains the generalized velocities of the shell, $\dot{q}_j(t)$, and any substructures, $\dot{q}_{\sigma j}(t)$, and is always produced during an execution of USLOB. The second file, called the "shell-substructure transformation file", is produced only when internal appendages are attached to the shell. This file contains the shell-substructure transformation matrices [the C_{ji} of Eq. (9)] needed to describe the motion of the shell, at the shell-substructure connection points, with reference to the coordinate systems used for the substructures. Further details of the USLOB code may be found in Appendix F.

e. Phase IV - Plots of Velocity-Time Histories

1. General Information

After the completion of Phases I-III of an ELSHOK calculation, it is usually necessary to present the computed results in the form of plotted velocity-time histories. The PUSLOB (PLOTS FOR USLOB) computer code is provided for this purpose. The input file, the generalized velocity file, and the shell-substructure transformation file, if applicable, must be available to complete Phase IV. All plotting using PUSLOB is performed on a TEKTRONIX graphics terminal (4010, 4051, 4052, e.g.) in conjunction with a TEKTRONIX hard copy unit (4631, e.g.). User provided input selects from among the various output options, specifies labeling, adjusts scales, and the like.

2. Summary of Output Options

Using PUSLOB, plots of velocity-time histories may be produced for any point on the shell S and for any nodal point on any internal appendage σ.

At the discretion of the user, velocity plots for the shell may be provided in the local coordinate system and/or in the global coordinate system. In the local system, the inward normal (n), meridional (s), and circumferential (θ) directions of Fig. 1 are employed. In the global system, the athwartship (-Z), fore-aft (+X), and downward (-Y) directions of Fig. 1 are used. All plotting for a substructure is performed with reference to the coordinate system of the given substructure. In addition to plotted results, coded card-image time histories may be prepared for transmittal to other computer installations or for additional post-processing.

If pseudo-velocity shock spectra are required, these may be produced from the card-image file mentioned above. At present, shock spectra are not produced by the PUSLOB code, but by a separate small computer program of the type available at most scientific establishments. Software for a TEKTRONIX 4051 or 4052 is available with ELSHOK to produce pseudo-velocity shock spectra from velocity-time histories, after processing the time histories on a digital tablet (digitizer). The inclusion of the spectra capability in PUSLOB is a modification planned for the future. Appendix G gives more information about the use of the PUSLOB computer code.

IV. SAMPLE PROBLEMS

a. Sample Problem 1

Consider the structure shown in Fig. 3 to be immersed, initially at rest, in an infinite expanse of water and to be excited by side-on loading in the form of a spherical step wave of constant strength P=10 psi (68.947 kN/m^2). The acoustic wave is assumed to emanate from a point located opposite the longitudinal midpoint of the structure at a stand-off distance of 69.9 ft (21.305 m). Moveover, the incident wave is assumed to impinge upon the submerged structure so as to excite doubly symmetrical motion (with respect to the incident wave). The orientation of the shock wave may be seen in Fig. 3. This sample problem involves only a full model of the shell S and illustrates the use of the straightforward modeling technique discussed in Appendix A. One substructure σ is included in the problem, namely, the long beam and its support brackets (Details 1 and 2 of Fig. 3). The bars and mounting plates (Details 3 and 4, respectively, of Fig. 3) are regarded as concentrated masses rigidly attached to the shell S.

The symmetrical in-vacuo mode shapes and corresponding natural frequencies of the empty shell of revolution S were obtained by applying the BOSOR4 computer program to the two-segment mathematical model pictured in Fig. 4. For this computation, the numerous light stiffeners were treated in the orthotropic approximation, while the heavy stiffeners were treated as discrete rings. The end closure of Fig. 4 was modeled as a flat plate. Only families of modes having circumferential wave numbers N=0,1,2,3 were considered, and 30 modes were computed for each of these values of N. Less than 3 min of computer time

(central processor time) was used on a CDC 6600 for these calculations. Elimination of the torsional and rolling modes for N=0 reduced the number of modes of S employed in the solution to 115. Appendix A contains a listing of the input data prepared for the BOSOR4 code for Sample Problem 1.

The virtual mass applicable to doubly symmetrical response of the submerged structure of Fig. 3 was computed using the ACESNID computer code. For the current problem, the cavity in the fluid was represented by 12 annular source bands on each end closure and 75 source bands on the cylindrical section. In performing the calculation of the virtual mass, the speed of sound in the fluid was taken to be 5000 ft/sec (1524 m/sec), corresponding to a diameter transit time of 0.56 msec. For each circumferential harmonic N considered, the normal motion of the wet surface of the shell S was described by 15 surface expansion functions corresponding to Eq. (13) for the cylindrical section and 3 surface expansion functions corresponding to Eqs. (14) and (15) for the end closures. Thus, 72 surface expansion functions in all were employed in the solution. About 8 min of central processor time was required on a CDC 6600 to determine the virtual mass. The input data for the virtual mass calculation of Sample Problem 1 may be found in Appendix B.

The symmetrical in-vacuo mode shapes and corresponding natural frequencies of the substructure σ were obtained by applying the SAPIV computer code to the assemblage of finite elements displayed in Fig. 5. For this calculation, 24 beam elements were used to connect 25 nodal points, and the substructure σ was constrained to move in the X - Z (horizontal) plane because of the doubly symmetrical nature of the problem under study.

It should be pointed out that an element release was employed to prevent the development of a bending moment in the support bracket at the junction of the beam and bracket and that the weight of the support bracket was neglected. The first 5 symmetrical fixed-based modes were computed and employed in the subsequent shock response calculation. About 3 sec of CDC 6600 computer time was required for the execution of the SAPIV code. The set of input data used for this calculation is listed in Appendix D.

The shell-fluid file (Appendix C) was produced by means of the PIFLASH computer program. Less than 30 sec of central processor time on a CDC 6600 was needed for this purpose. The set of input data prepared for PIFLASH for Sample Problem 1 may be found in Appendix C. Processing to produce the substructure file (Appendix E) for Sample Problem 1 was accomplished by means of the PICRUST computer code in less than 15 sec of CDC 6600 computer time. Appendix E contains the input data employed for this execution of PICRUST. The input file for the response calculation of Sample Problem 1 was constructed from the shell-fluid and substructure files above using the control statement procedure discussed in Appendix F. This procedure required less than 2 sec of central processor time on a CDC 6600.

Solution of the transient structure-fluid interaction problem described earlier was performed by means of the USLOB code. A constant time step of 0.000025 sec (about 1/25 of a diameter transit time) was used, and the calculation was continued for more than 8 diameter transit times. Nearly 8 min of CDC 6600 central processor time was required to compute the shock response of the submerged structure. Appendix E contains the input data employed for this calculation. Plotting of the results was accomplished by means of the PUSLOB computer code, about 15 sec of CDC 6600 computer time being required for the problem at hand.

The input data employed to produce the desired velocity-time plots may be found in Appendix G. Figures 6-13 display the results of the calculation performed as Sample Problem 1.*) These figures are embellished versions of those produced by the PUSLOB code.

b. Sample Problem 2

Consider the structure shown in Fig.3 to be immersed, initially at rest, in an infinite expanse of water and to be excited by side-on loading in the form of a spherical wave of decaying exponential profile in time. The shock wave is assumed to be produced by the underwater explosion of a 3 lb (1.36 kg) spherical charge of TNT at a point located opposite the longitudinal midpoint of the structure at a stand-off distance of 2.802 ft (0.854 m),or one model diameter. As in Sample Problem 1, the incident wave is assumed to impinge upon the submerged structure so as to excite doubly symmetrical motion. The orientation of the shock wave may be seen in Fig. 3. This sample problem is presented to demonstrate the use of the approximate modeling technique, involving a region of interest, discussed in Appendix A. One substructure σ is included in the problem, namely, the long beam and its support brackets (Details 1 and 2 of Fig. 3). The bars and mounting plates (Details 3 and 4, respectively, of Fig. 3) are regarded as concentrated masses rigidly attached to the shell S.

The symmetrical in-vacuo mode shapes and corresponding natural frequencies of the empty shell of revolution S were obtained by applying the BOSOR4 computer code to the three-segment full model and to the one - segment compartment model pictured in Fig. 14. As in Sample Problem 1,

*) The term "PROBONE" (from PROBLEM ONE) appearing in the alphameric headings of Figs. 6-13 is the name of the input file used for Sample Problem 1.

the numerous light stiffeners were treated in the orthotropic approximation, while the heavy stiffeners were treated as discrete rings. The end closure of Fig. 14 was modeled as a flat plate. Only families of modes having circumferential wave numbers N=0,1,2,3 were considered in the modal analysis of the shell S. For N=1, 30 modes were computed using the full model of S. For each value of N=0, 2,3, 20 modes were determined using the compartment model of S. Less than 2 min of computer time (central processor time) was used on a CDC 6600 for these calculations. The use of 15 breathing modes for N=0, 15 modes for N=2, 15 modes for N=3, and 30 modes for N=1 brought the total number of modes of S employed in the transient response problem to 75. Appendix A contains the input data prepared for the BOSOR4 code for Sample Problem 2.

The virtual mass suitable for use with the approximate modeling technique employed for the shell S was computed using the ACESNID computer code. The cavity in the fluid was represented by 10 annular source bands on each end closure and 75 source bands on the cylindrical section. In performing the calculation of the virtual mass, the speed of sound in the fluid was taken to be 5000 ft/sec (1524 m/sec),corresponding to a diameter transit time of 0.56 msec. For N=1, the normal motion of the wet surface of the shell S was described by 8 surface expansion functions corresponding to Eq. (13) for the cylindrical section, 3 surface expansion functions corresponding to Eqs. (14) and (15) for the end closures, and 7 surface expansion functions corresponding to Eq. (16) for the compartment region. For each value of N=0,2,3, 7 surface expansion functions corresponding to Eq. (16) were used for the compartment region. Thus, 39 surface expansion functions in all were employed in the solution. About

7 min of central processor time was required on a CDC 6600 to determine the virutal mass. The input data for the virtual mass calculation of Sample Problem 2 may be found in Appendix B.

The symmetrical in-vacuo fixed-base modes and corresponding natural frequencies of the substructure σ were obtained by applying the SAPIV computer code to the assemblage of finite elements displayed in Fig. 5. Since the identical mathematical model was used for Sample Problem 1, the discussion of the modeling and computational details will not be repeated here. It should be pointed out that, for Sample Problem 2, the shell cross section in the note of Fig. 5 (B) must be changed to mesh point 1 of segment 3. The set of input data used for the modal analysis of the substructure σ for Sample Problem 2 is listed in Appendix D.

The PIFLASH computer program was utilized to produce the shell-fluid file. Less than 15 sec of central processor time on a CDC 6600 was needed for this purpose. The set of input data prepared for PIFLASH for Sample Problem 2 may be found in Appendix C. Processing to produce the substructure file for Sample Problem 2 was accomplished by means of the PICRUST computer code in less than 15 sec of CDC 6600 computer time. Appendix E contains the input data employed for this execution of PICRUST. The input file for the response calculation of Sample Problem 2 was constructed from the shell-fluid and substructure files above using the control statement procedure discussed in Appendix F. This procedure required less than 2 sec of central processor time on a CDC 6600.

Solution of the transient structure-fluid interaction problem described earlier was performed by means of the USLOB code. A constant time step of 0.000025 sec (about 1/25 of a diameter transit time) was used,

and the calculation was continued for more than 8 diameter transit times. Nearly 8 min of CDC 6600 central processor time was required to compute the shock response of the submerged structure. Appendix E contains the input data employed for this calculation. Plotting of the results was accomplished by means of the PUSLOB computer code, about 10 sec of CDC 6600 computer time being required for the problem at hand. The input data employed to produce the desired velocity-time plots may be found in Appendix G. Figures 15-19 display the results of the calculation performed as Sample Problem 2.*) These figures are embellished versions of those produced by the PUSLOB code.

*) The term "PROBTWO" (from PROBLEM TWO) appearing in the alphameric headings of Figs. 15-19 is the name of the input file used for Sample Problem 2.

REFERENCES

[1] R. Atkatsh and R.P. Daddazio, "Dynamic Elasto-Plastic Response of Shells in an Acoustic Medium - User's Manual for the EPSA Code", Technical Report No. 27, Office of Naval Research, Contract No. N00014-78-C-0820, Weidlinger Associates, New York, New York, March 1980.

[2] D. Ranlet, F.L. DiMaggio, H.H. Bleich and M.L. Baron, "Elastic Response of Submerged Shells with Internally Attached Structures to Shock Loading", International Journal of Computers and Structures, Vol. 7, No. 3, June 1977, pp. 355-364.

[3] D. Ranlet and F.L. DiMaggio, "Transient Response of Shells with Internally Attached Structures", International Journal of Computers and Structures, Vol. 9, No. 5, November 1978, pp. 475-481.

[4] T.L. Geers, "Residual Potential and Approximate Methods for Three-Dimensional Fluid-Structure Interaction Problems", Journal of the Acoustical Society of America, Vol. 49, No. 5, (Part 2), 1971, pp. 1505-1510.

[5] R. Courant and D. Hilbert, "Methods of Mathematical Physics", Vol. I, Interscience Publishers, Inc., New York, New York, 1966, pp. 50-51.

[6] R.H. Cole, "Underwater Explosions", Princeton University Press, Princeton, New Jersey, 1948, pp. 228-231.

[7] D. Bushnell, "Stress, Stability and Vibration of Complex Branched Shells of Revolution: Analysis and User's Manual for BOSOR4", LMSC-D243605, Lockheed Missiles and Space Co., Inc., Sunnyvale, California, March 1972.

[8] K.J. Bathe, E.L. Wilson and F.E. Peterson, "SAPIV - A Structural Analysis Program for Static and Dynamic Response of Linear Systems", Earthquake Engineering Research Center, Report No. EERC 73-11, College of Engineering, University of California, Berkeley, California, April 1974.

[9] "UPDATE Reference Manual", Publication No. 60342500, Revision F, Control Data Corporation, Publication and Graphics Division, Arden Hills, Minnesota, June 1975.

[10] H. Benaroya and D. Ranlet, "Accession to Inertia and Damping for Structures in an Infinite Acoustic Medium - Theoretical and User's Guide for ACESION and ACESNID", Defense Nuclear Agency, Contract No. DNA 001-81-C-0048, Weidlinger Associates, New York, New York, December 1981, in preparation.

[11] A. Ralston and H.S, Wilf, "Mathematical Methods for Digital Computers", Vol. I, John Wiley and Sons, Inc., New York, 1967, pp. 110-120.

[12] T. Simpson, "Mathematical Dissertations", London, 1743, p. 109.

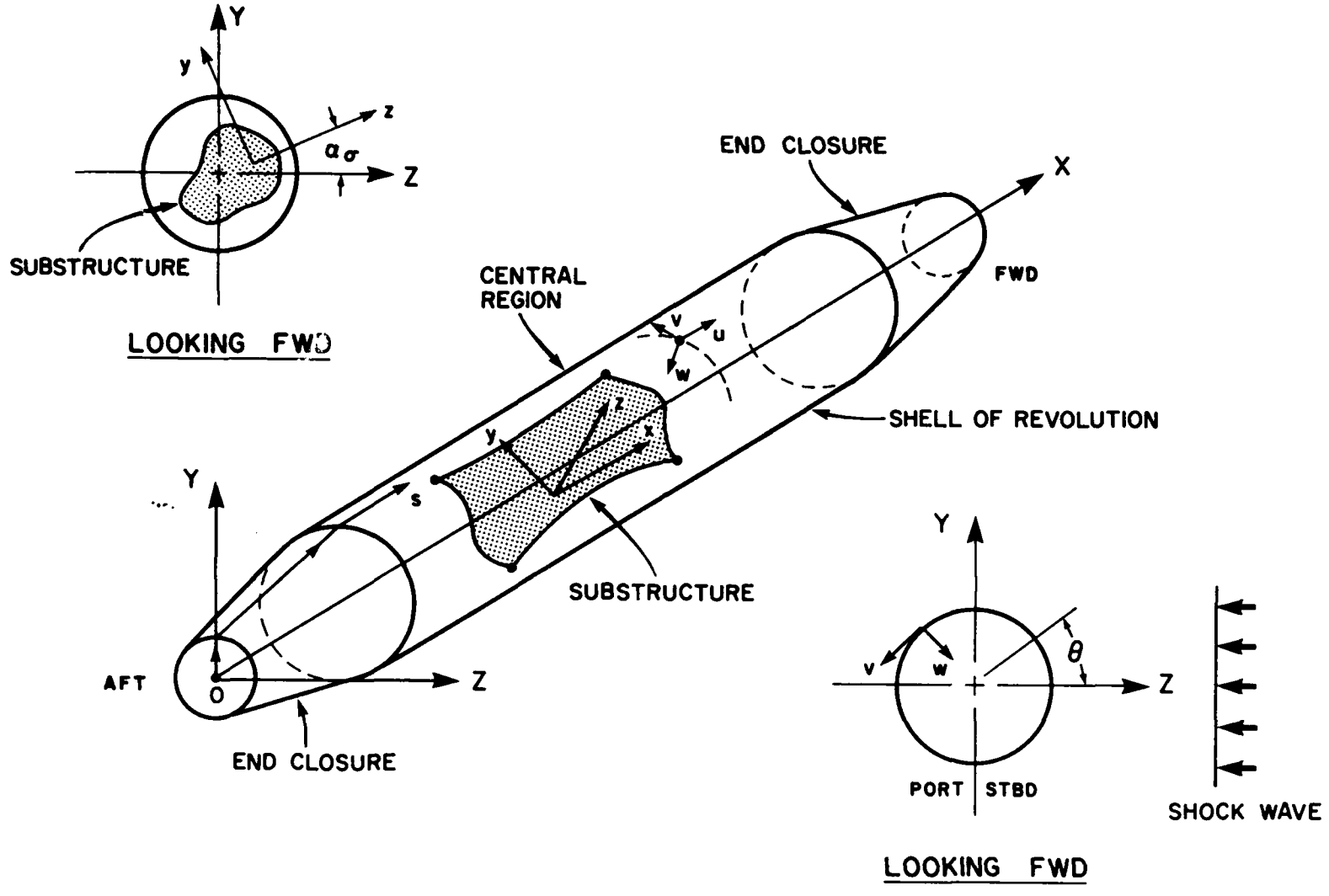


FIG. 1 TYPICAL SHELL-SUBSTRUCTURE CONFIGURATION

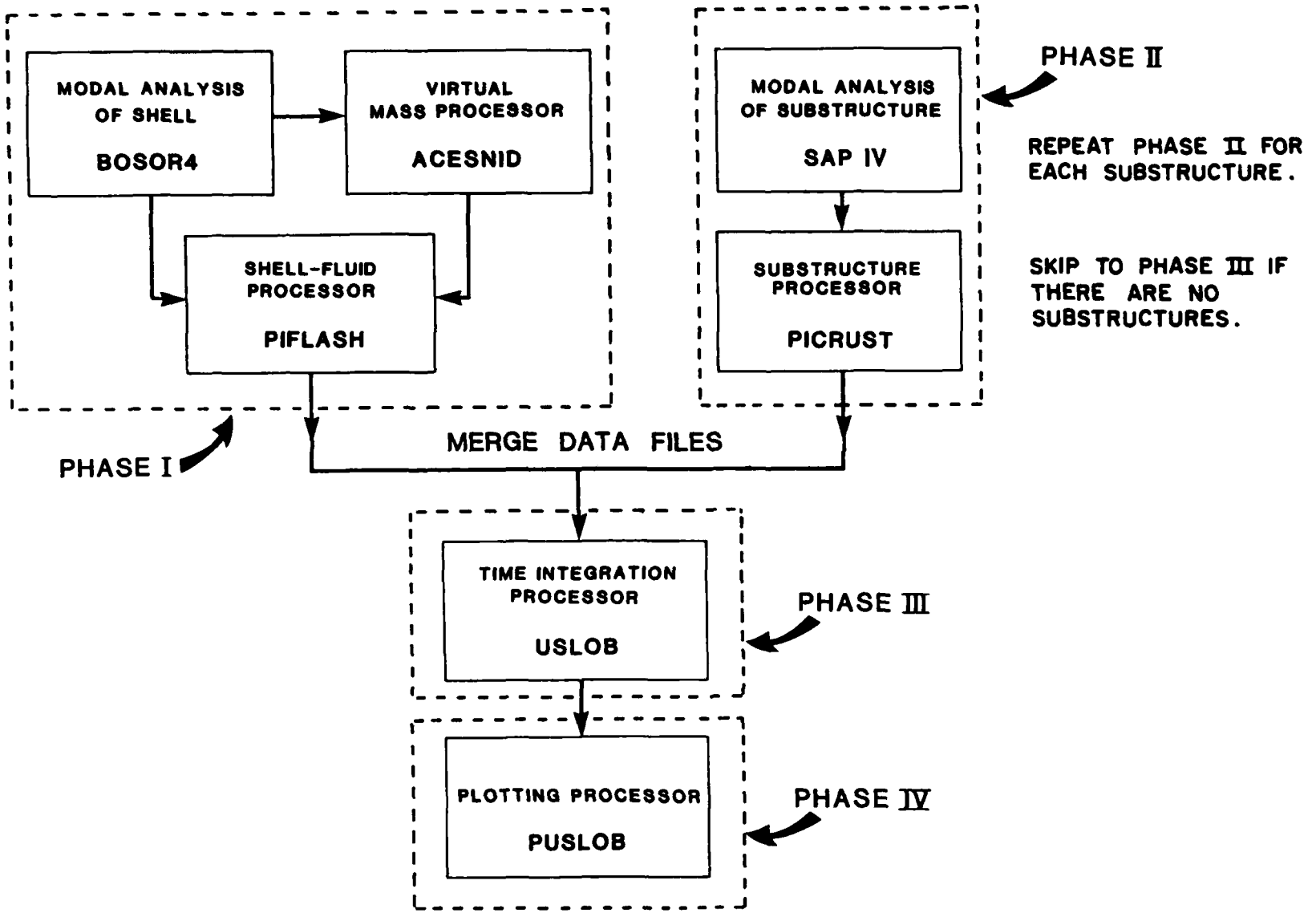
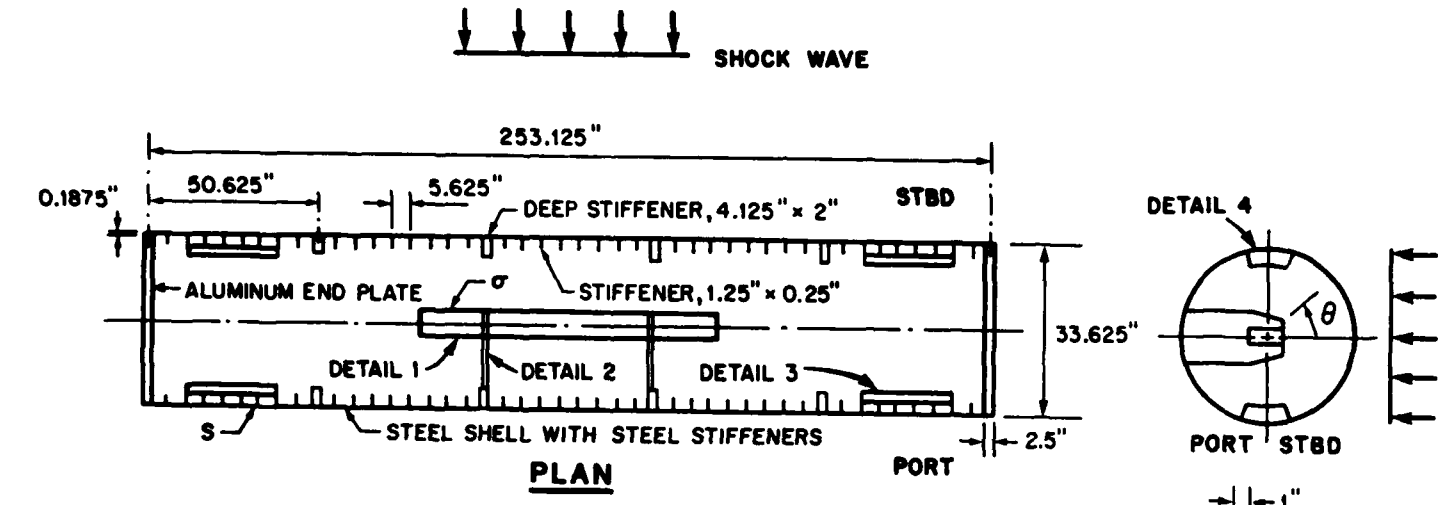
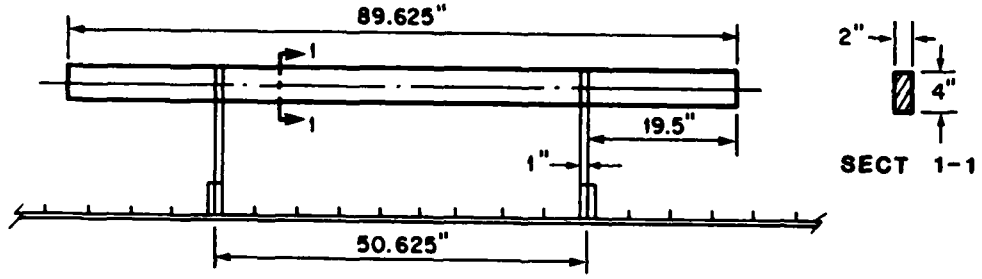
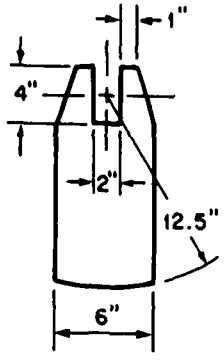


FIG. 2 ORGANIZATION OF THE ELSHOK CODE

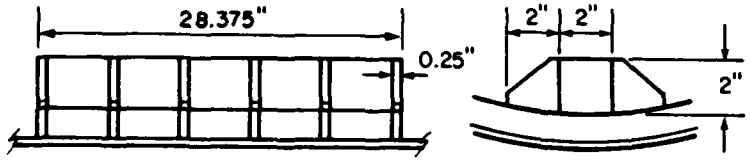




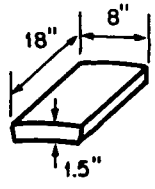


DETAIL 1: STEEL BEAM WITH STEEL BRACKETS (SUBSTRUCTURE σ)

DETAIL 2: STEEL BRACKET

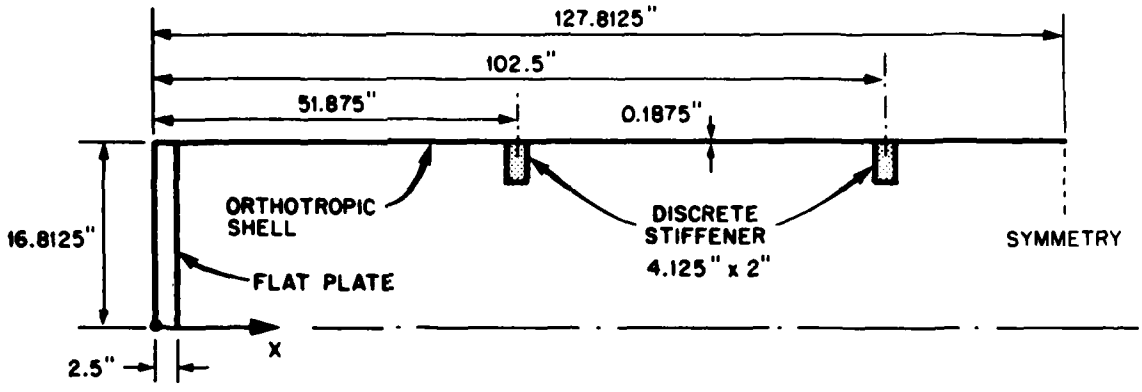


DETAIL 3: STEEL BAR

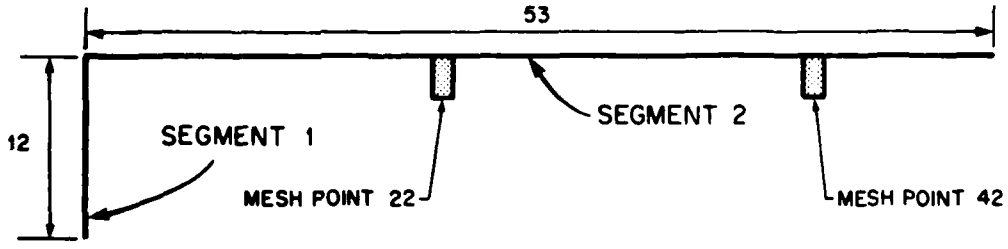


DETAIL 4:
STEEL MOUNTING PLATE

FIG. 3 STRUCTURAL MODEL -- SAMPLE PROBLEM 1



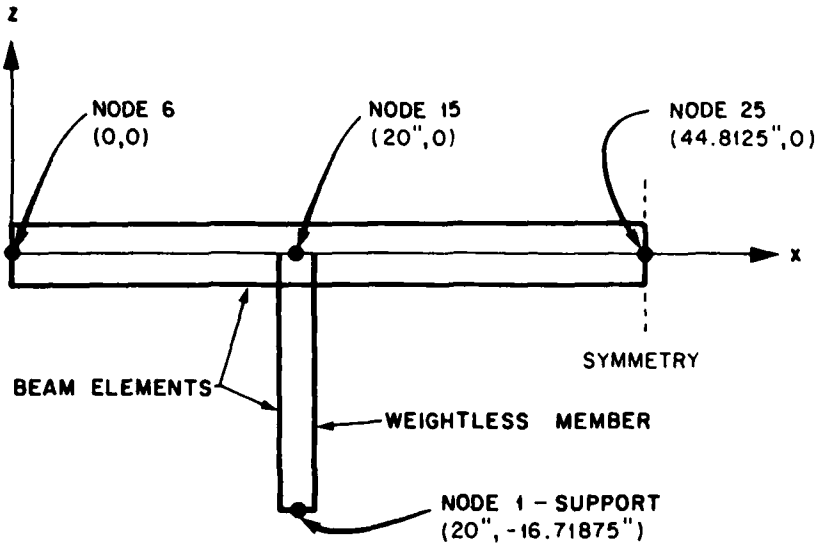
(A) GEOMETRY



NOTE: THE BOSOR4 CODE EMPLOYS TWO MORE MESH POINTS IN EACH SEGMENT THAN THE NUMBER PROVIDED AS INPUT.

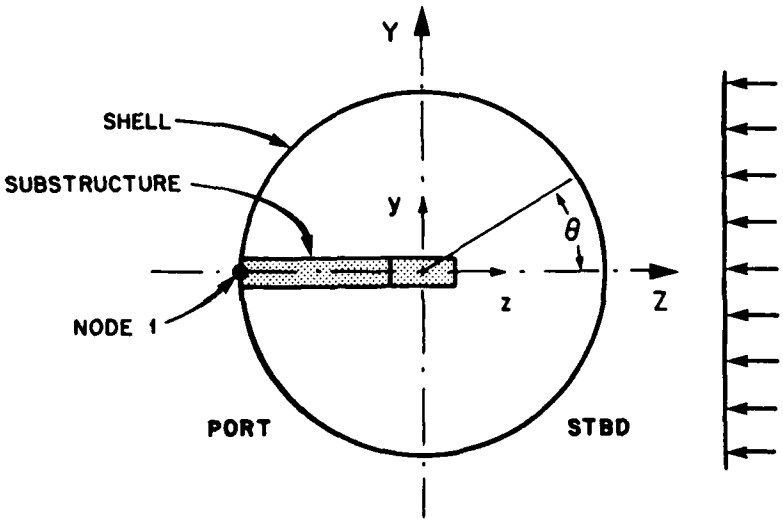
(B) DISTRIBUTION OF MESH POINTS ALONG REFERENCE MERIDIAN

FIG. 4 SHELL OF REVOLUTION SAMPLE PROBLEM 1



NOTE: ACTIVE PHYSICAL DEGREES OF FREEDOM ARE z - COMPONENTS OF TRANSLATION AND y - COMPONENTS OF ROTATION.

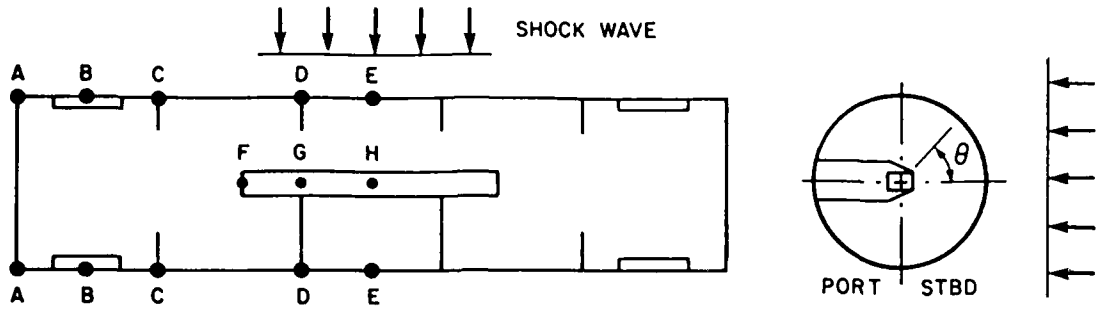
(A) FINITE ELEMENT MODEL OF SUBSTRUCTURE



NOTE: SHELL CROSS SECTION IS FOR MESH POINT 42 OF SEGMENT 2.

(B) SHELL-SUBSTRUCTURE CONNECTIVITY

FIG. 5 INTERNAL STRUCTURE SAMPLE PROBLEM 1



SAMPLE PROBLEM 1 (PROBONE) V-ATH 2/01

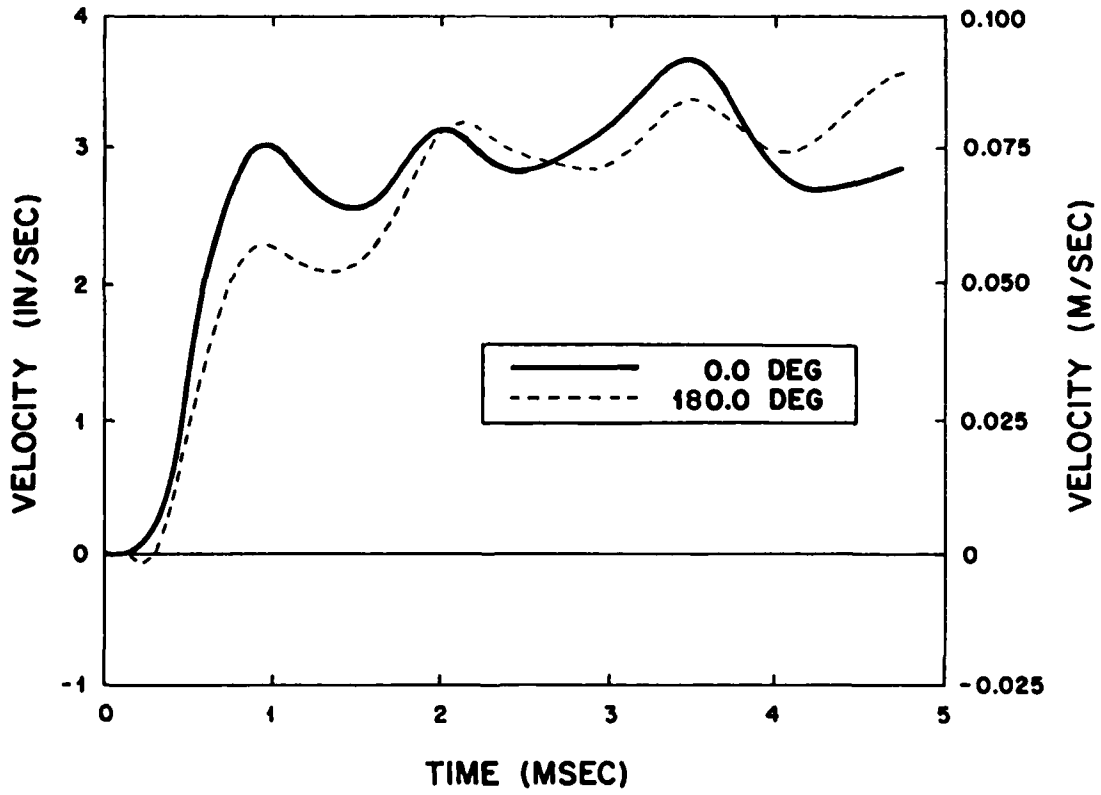
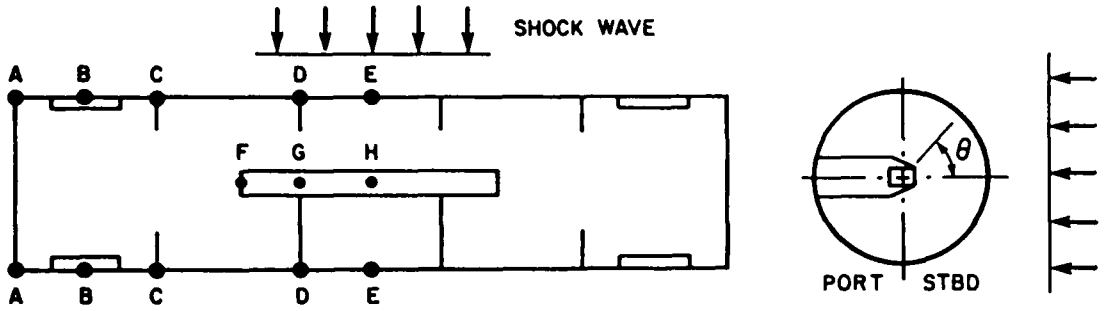


FIG. 6 ATHWARTSHIP VELOCITIES AT STATION A (SEGMENT 2, POINT 1) FOR SAMPLE PROBLEM 1



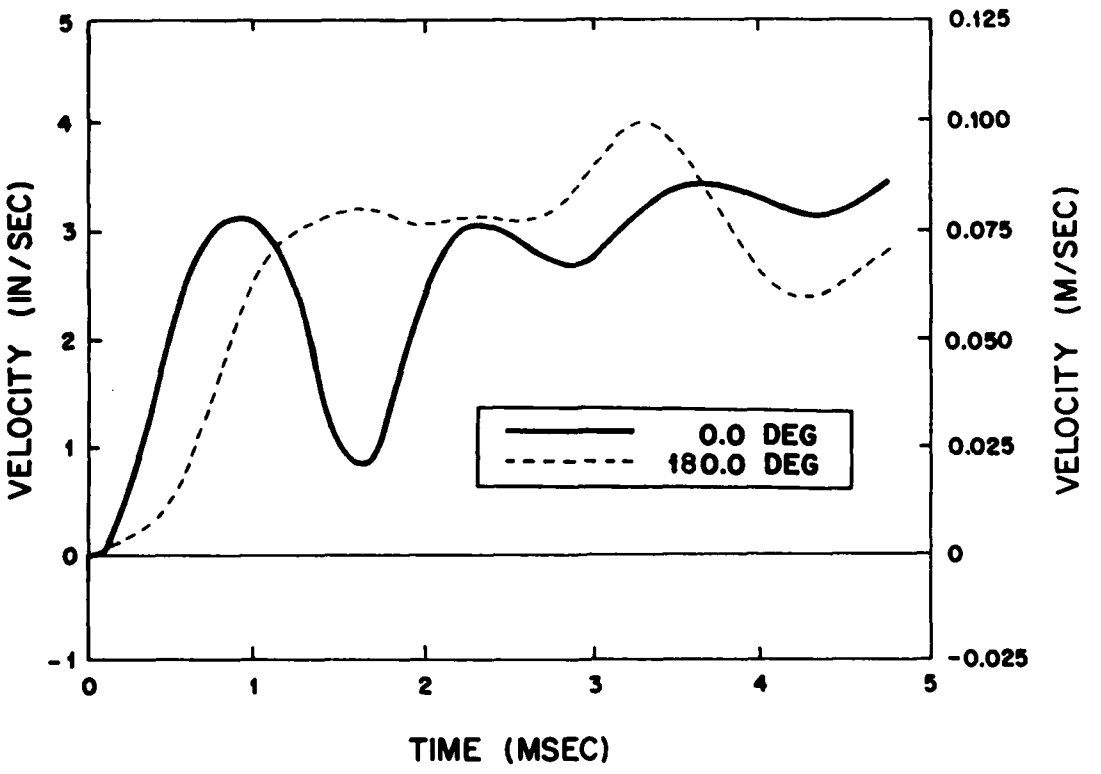
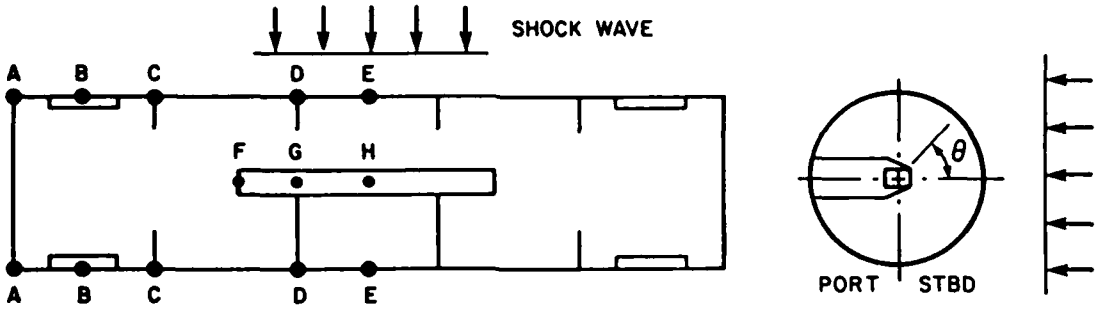


FIG. 7 ATHWARTSHIP VELOCITIES AT STATION B (SEGMENT 2, POINT 12) FOR SAMPLE PROBLEM 1



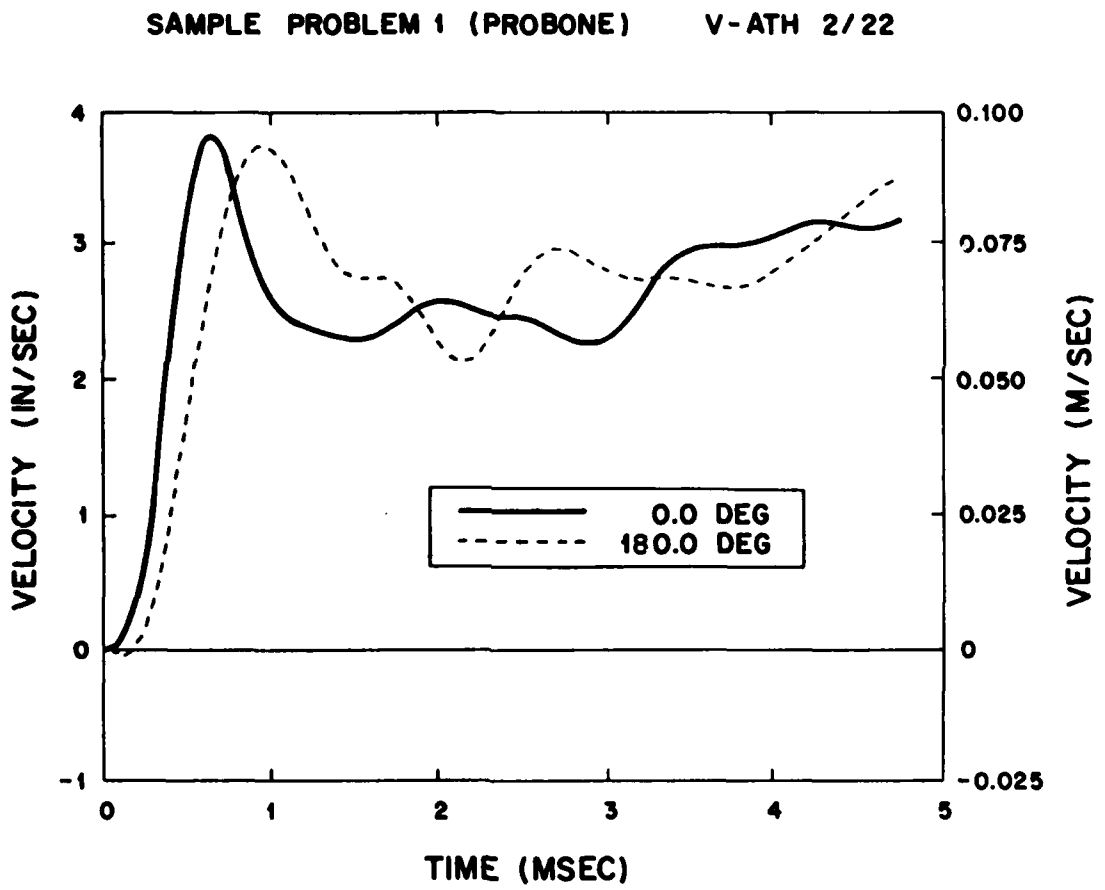
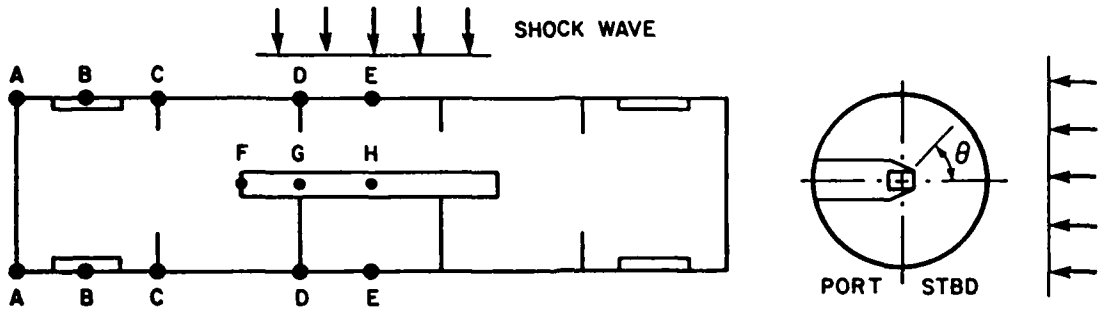


FIG. 8 ATHWARTSHIP VELOCITIES AT STATION C (SEGMENT 2, POINT 22) FOR SAMPLE PROBLEM 1



SAMPLE PROBLEM 1 (PROBONE) V-ATH 2/42

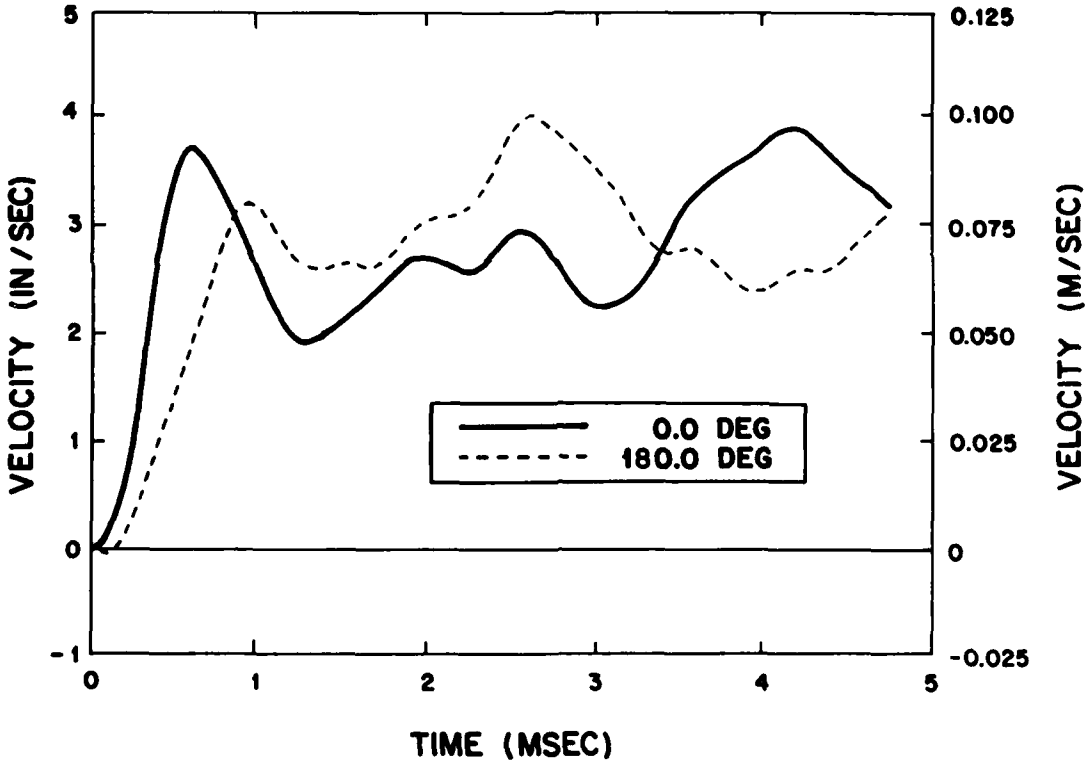
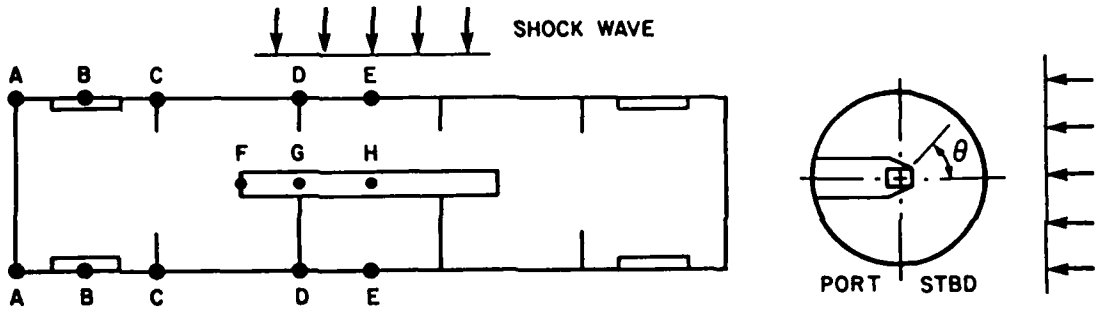


FIG. 9 ATHWARTSHIP VELOCITIES AT STATION D (SEGMENT 2, POINT 42) FOR SAMPLE PROBLEM 1



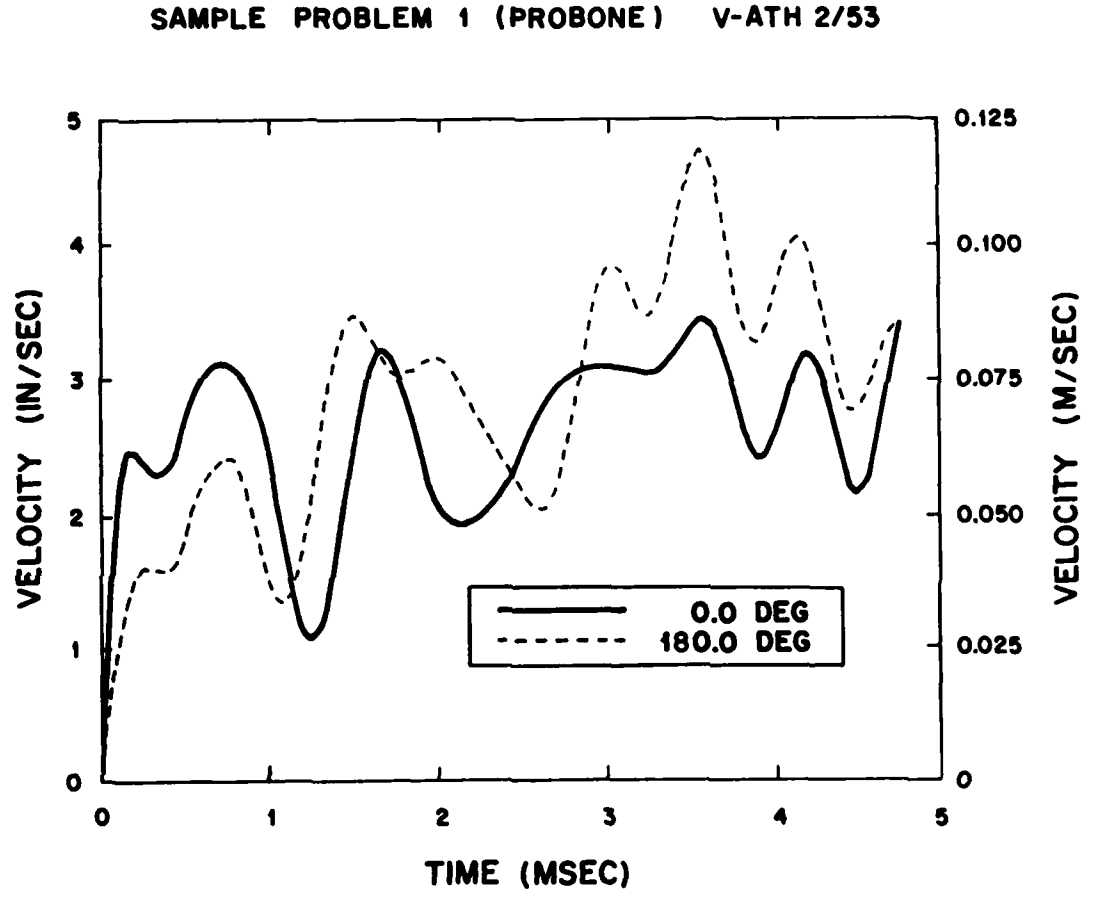
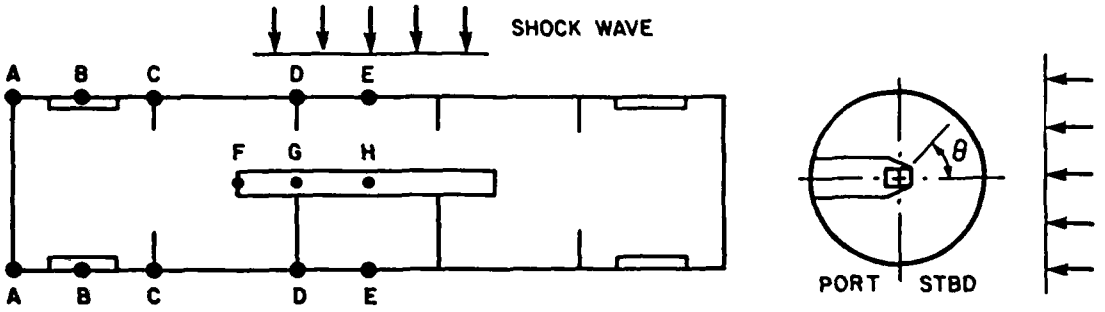


FIG. 10 ATHWARTSHIP VELOCITIES AT STATION E (SEGMENT 2, POINT 53) FOR SAMPLE PROBLEM 1



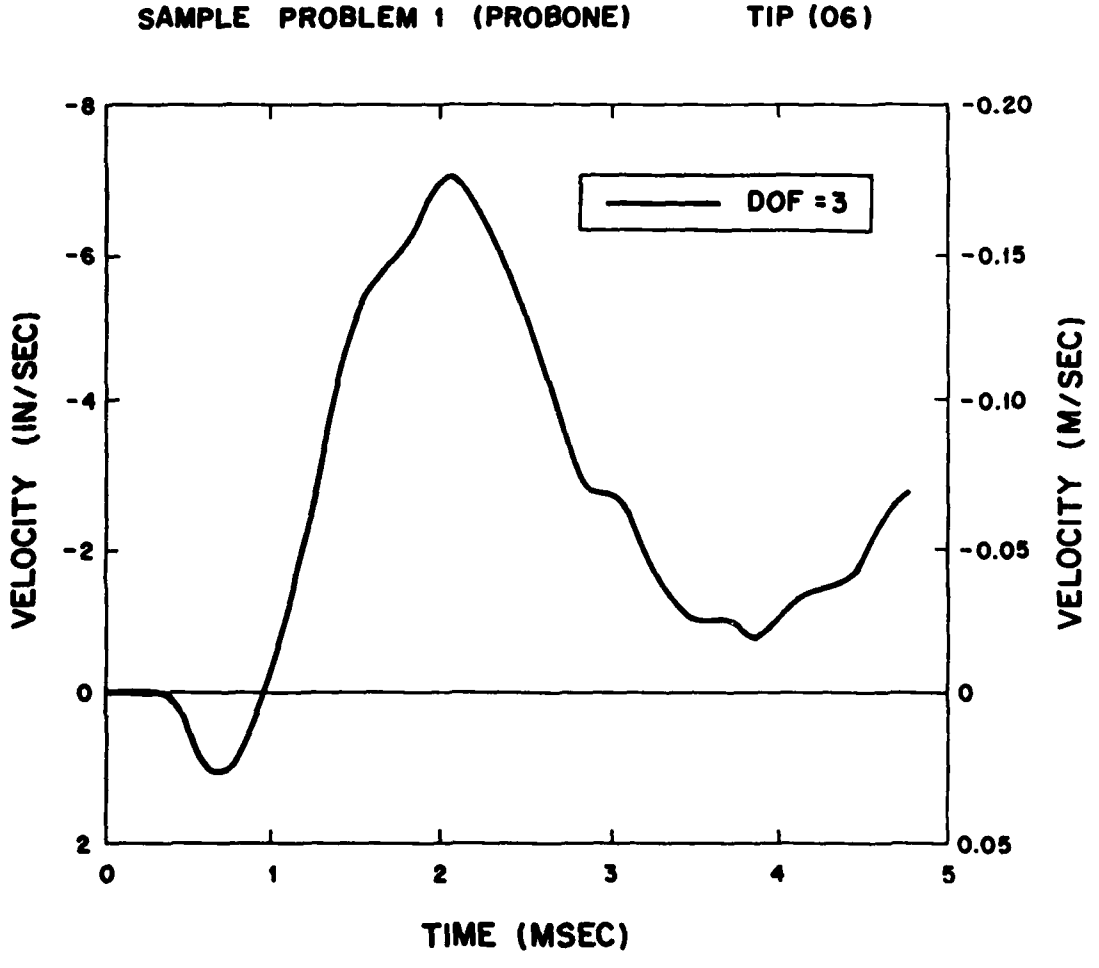
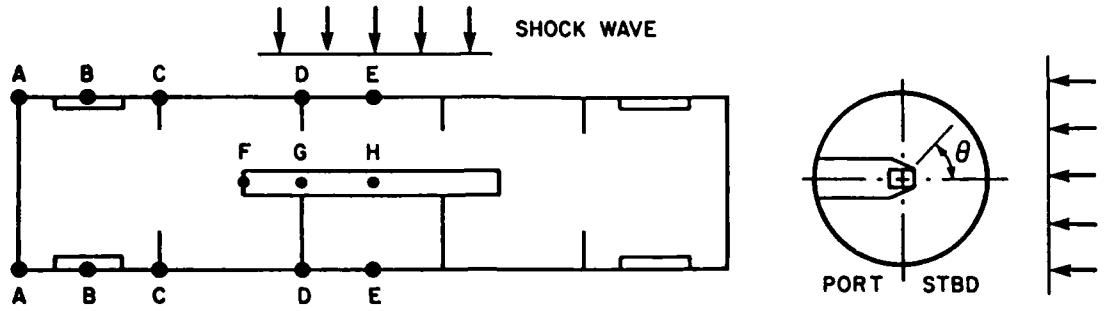


FIG. 11 HORIZONTAL (z) VELOCITY AT BEAM STATION F (NODE 6) FOR SAMPLE PROBLEM 1



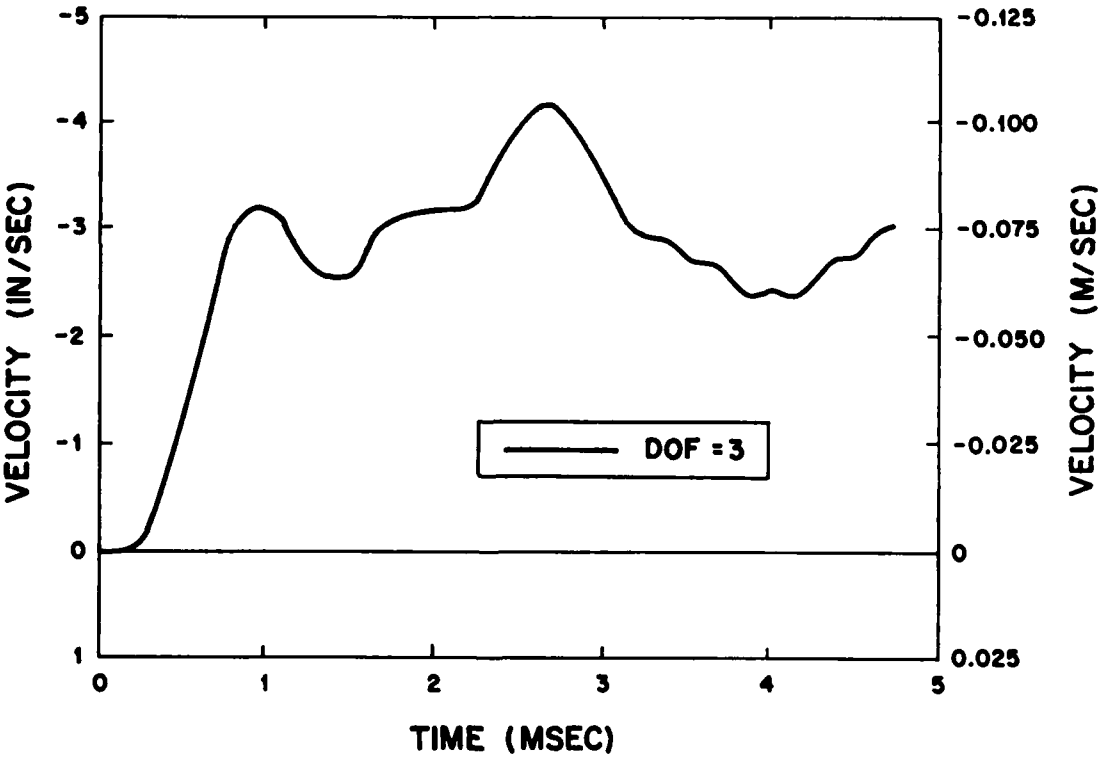
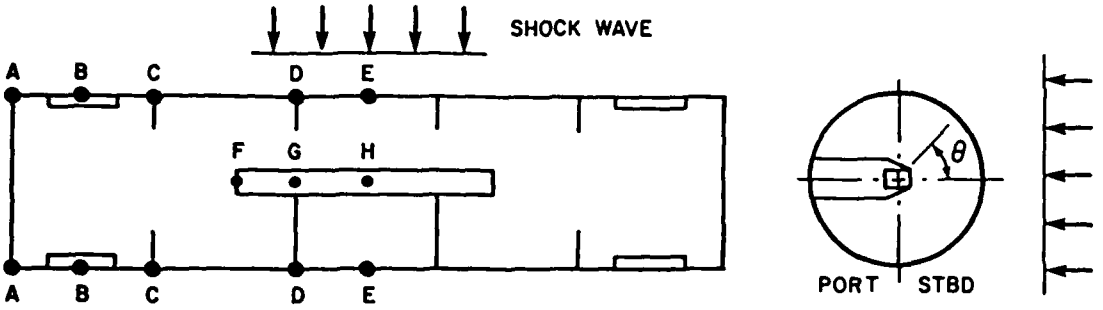


FIG. 12 HORIZONTAL (z) VELOCITY AT BEAM STATION G (NODE 15) FOR SAMPLE PROBLEM 1



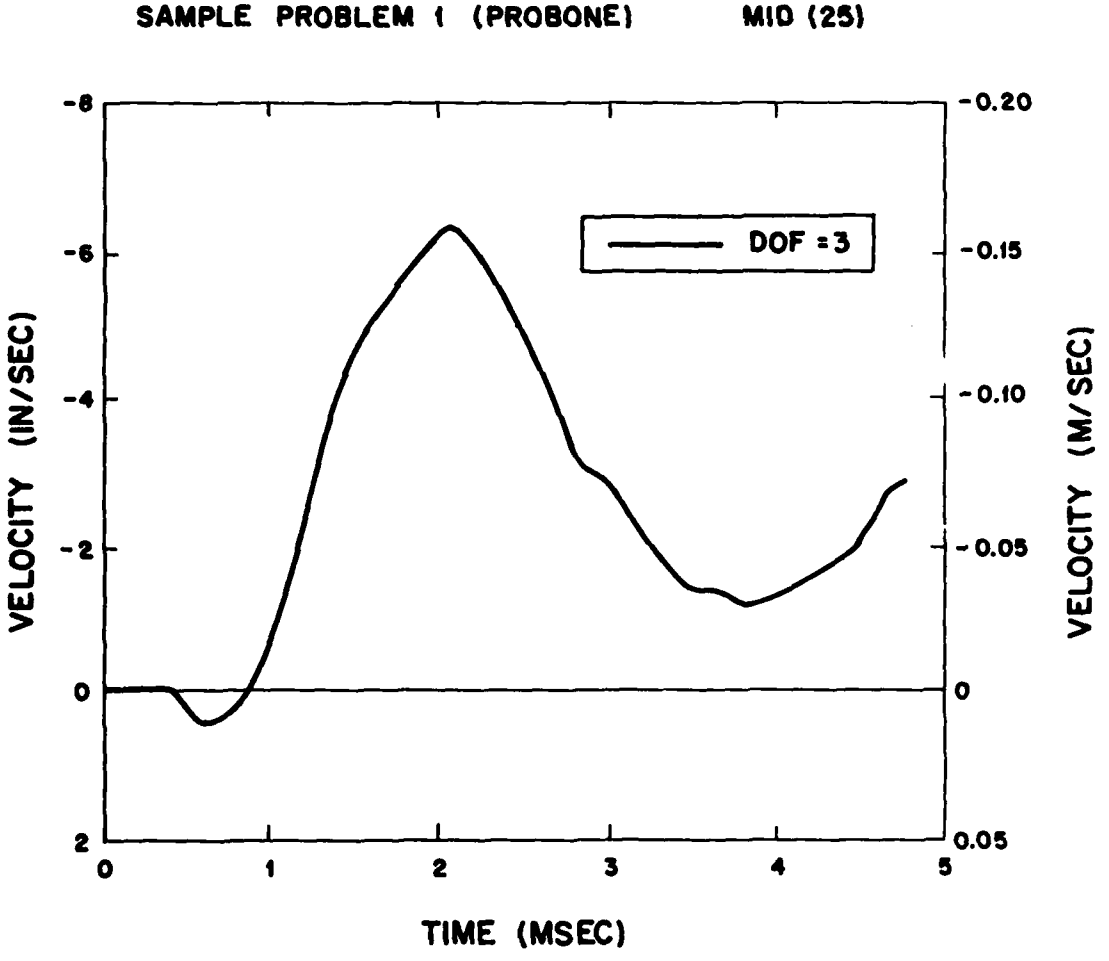
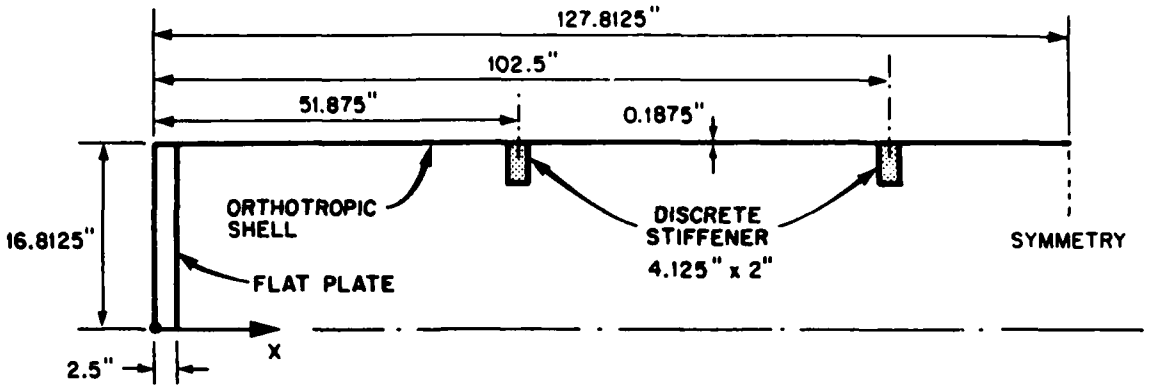
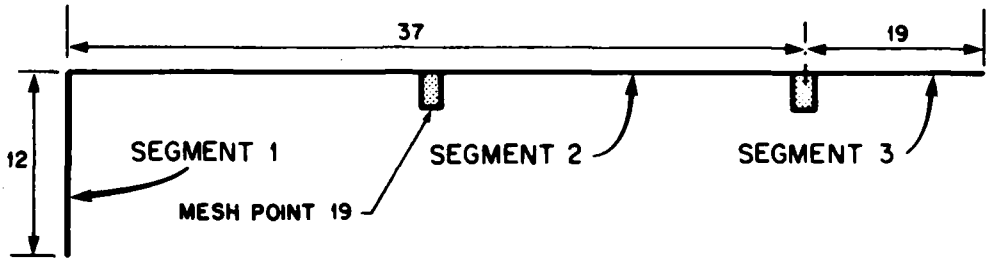


FIG. 13 HORIZONTAL (z) VELOCITY AT BEAM STATION H (NODE 25) FOR SAMPLE PROBLEM 1



(A) GEOMETRY OF FULL MODEL

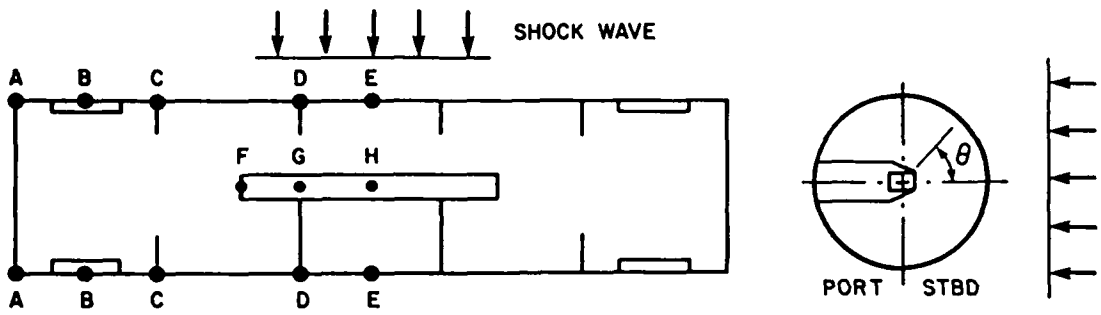


NOTE: THE BOSOR4 CODE EMPLOYS TWO MORE MESH POINTS IN EACH SEGMENT THAN THE NUMBER PROVIDED AS INPUT.

(B) DISTRIBUTION OF MESH POINTS ALONG REFERENCE MERIDIAN OF FULL MODEL

(C) GEOMETRY AND DISTRIBUTION OF MESH POINTS USED FOR COMPARTMENT MODEL

FIG. 14 SHELL OF REVOLUTION SAMPLE PROBLEM 2



SAMPLE PROBLEM 2 (PROBTWO) V-ATH 3/1

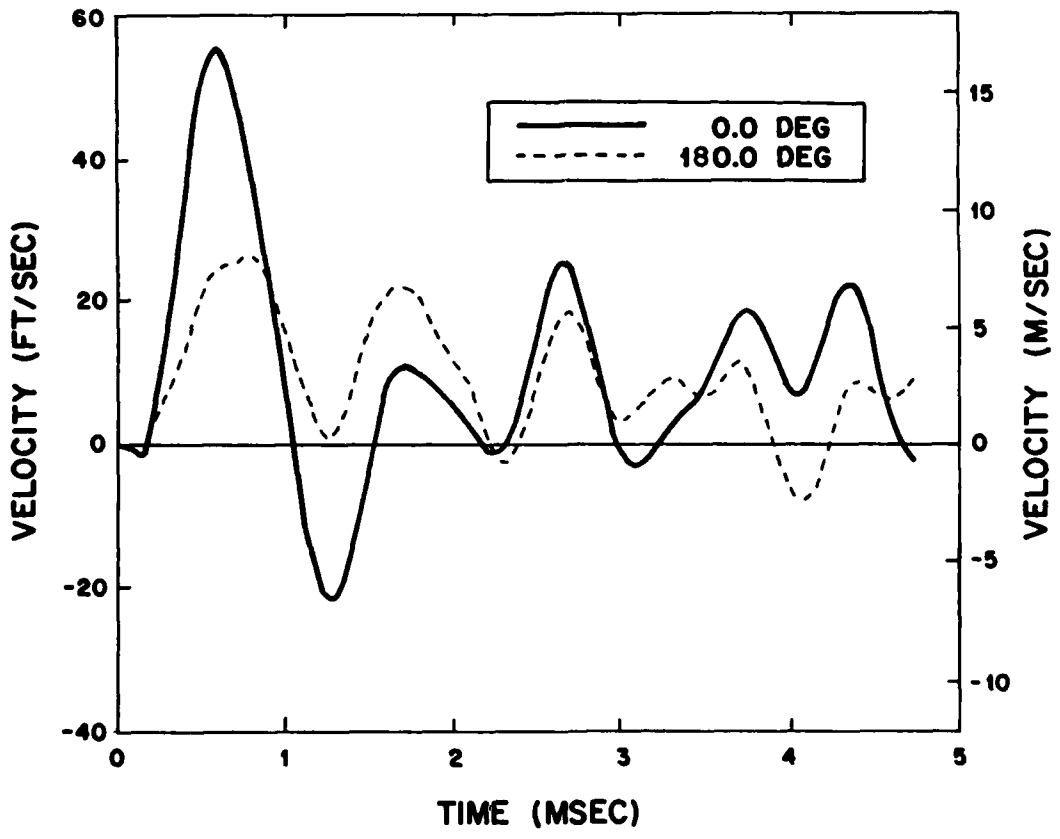
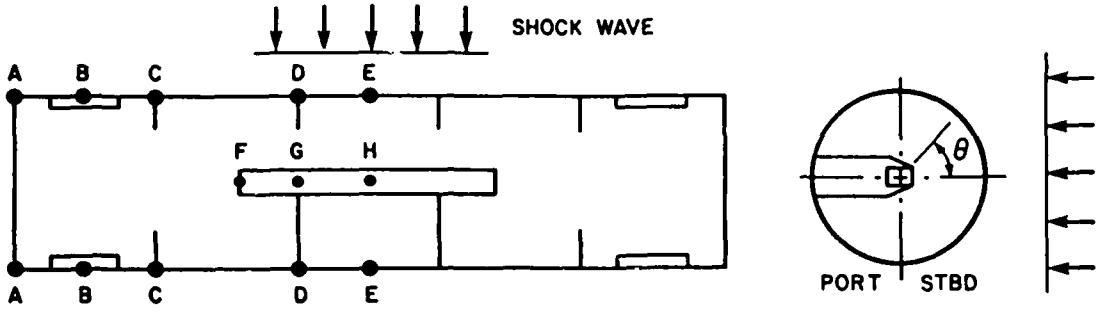


FIG. 15 ATHWARTSHIP VELOCITIES AT STATION D (SEGMENT 3, POINT 1) FOR SAMPLE PROBLEM 2



SAMPLE PROBLEM 2 (PROBTWO) V-ATH 3/19

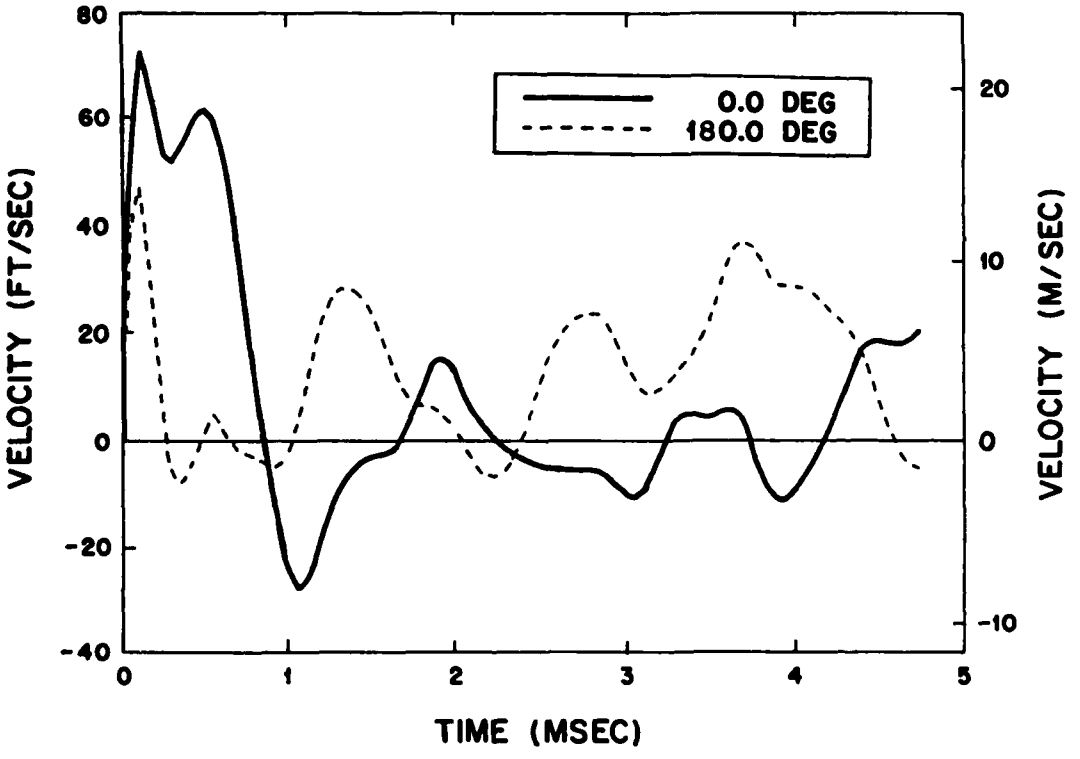
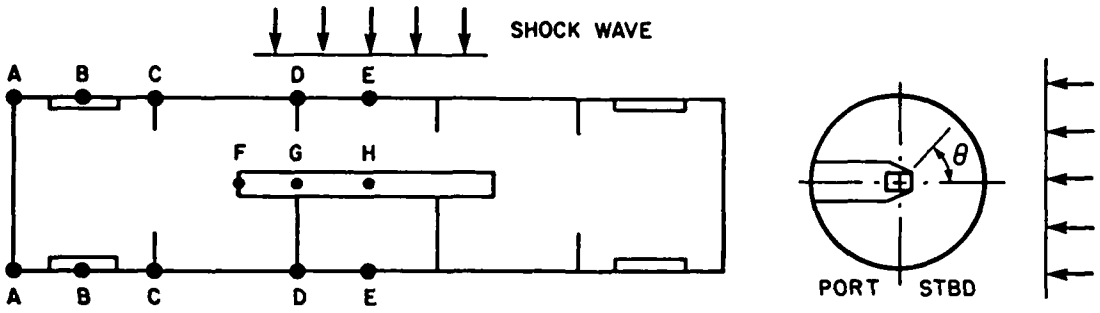


FIG. 16 ATHWARTSHIP VELOCITIES AT STATION E (SEGMENT 3, POINT 19) FOR SAMPLE PROBLEM 2



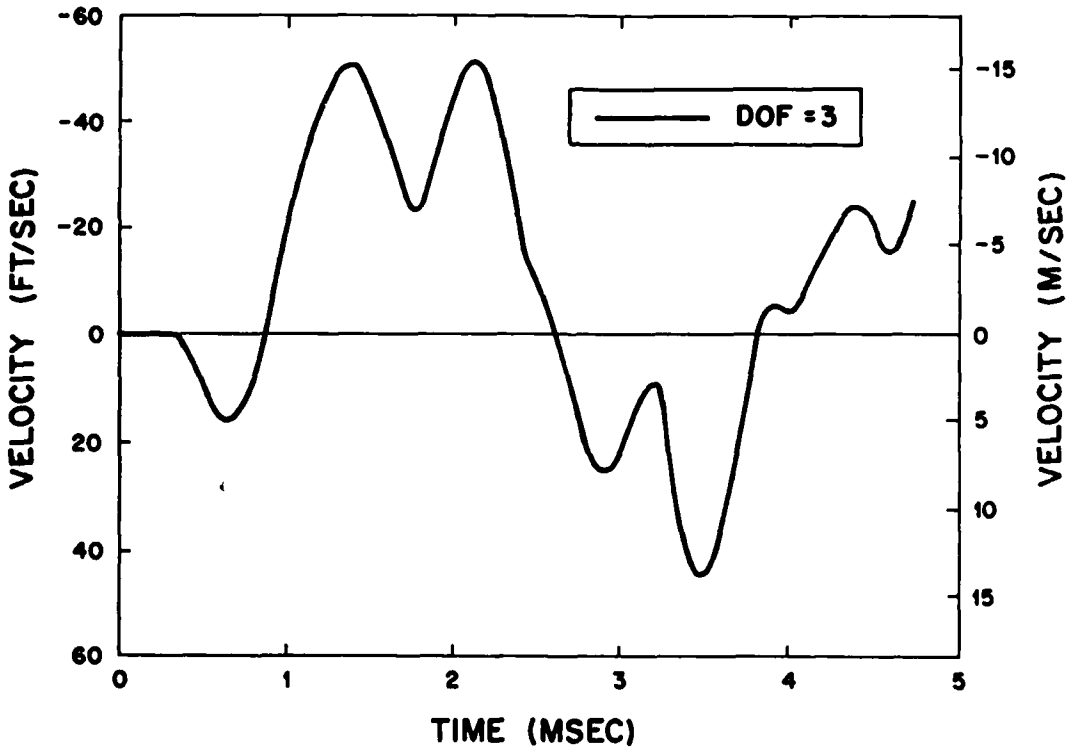
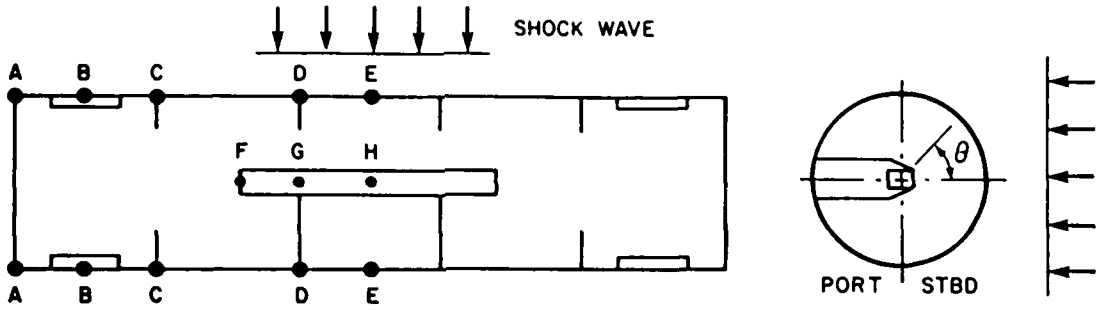


FIG. 17 HORIZONTAL (z) VELOCITY AT BEAM STATION F (NODE 6) FOR SAMPLE PROBLEM 2



SAMPLE PROBLEM 2 (PROBTWO) JCN(15)

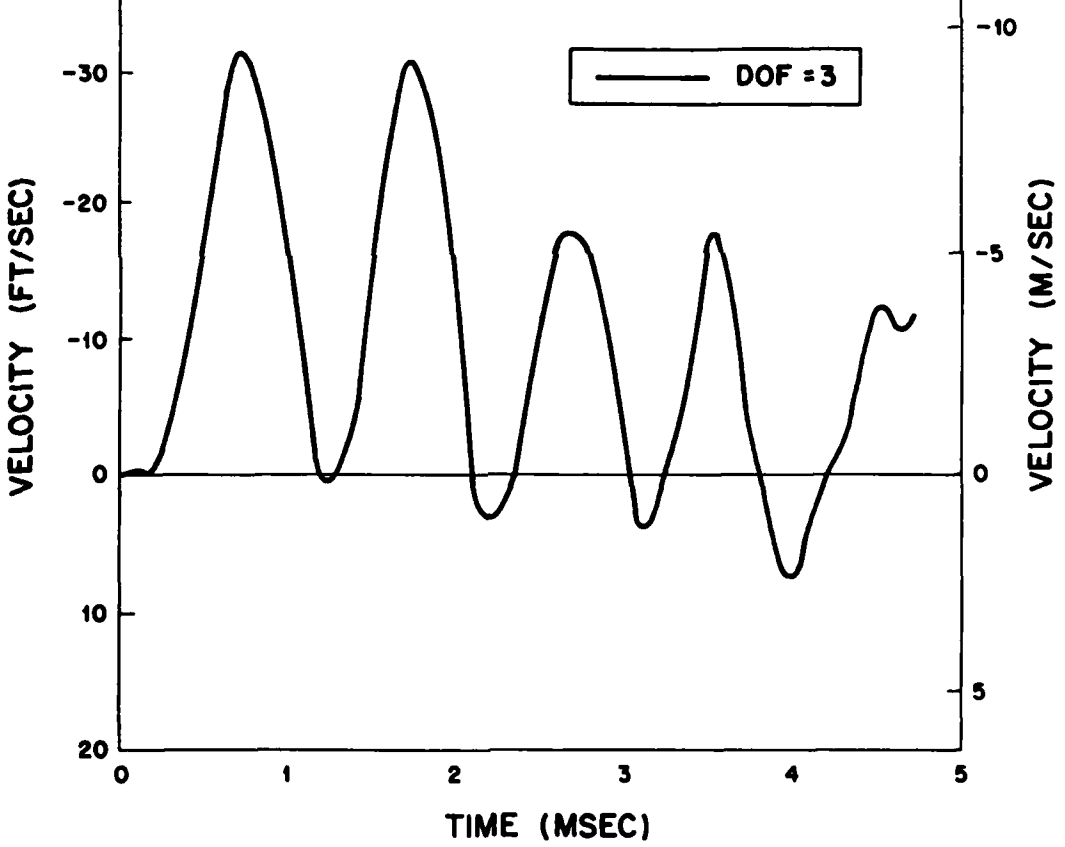
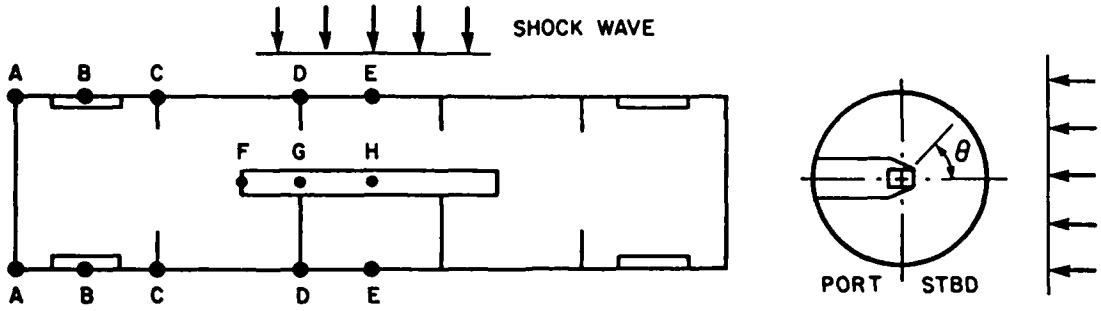


FIG. 18 HORIZONTAL (z) VELOCITY AT BEAM STATION G (NODE 15) FOR SAMPLE PROBLEM 2



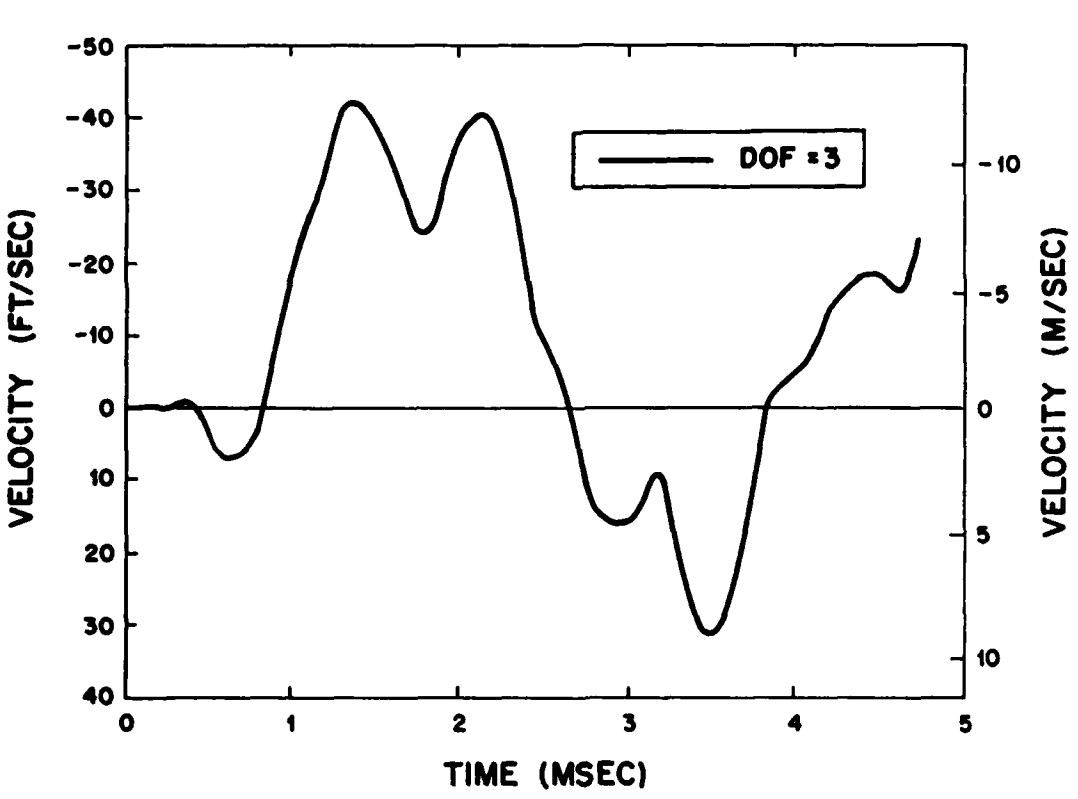


FIG. 19 HORIZONTAL (z) VELOCITY AT BEAM STATION H (NODE 25) FOR SAMPLE PROBLEM 2

APPENDIX A - THE USE OF BOSOR4

1. Background

The BOSOR4 computer code of Ref. [7] is included in the ELSHOK suite of computer programs for the modal analysis of the main body S. This code is quite general and applies to segmented, ring-stiffened, branched shells of revolution having various meridional geometries, wall constructions, and ring reinforcements. Comprehensive documentation of BOSOR4 may be found in Ref. [7]. A knowledge of the information given in Sections II and III of the present report will aid in the understanding of this appendix.

In the formulation employed in BOSOR4, the independent variables of the analysis are the meridional arc length s, measured along the shell reference meridian and the circumferential coordinate angle θ (Fig. 1). The dependent variables are the meridional (u), circumferential (v), and normal (w) components of displacement of the shell reference surface. The numerical analysis in BOSOR4 is based upon a finite difference energy method, so that in the computer program, the dependent variables are the displacement components at discrete points (mesh points) on the shell reference surface. Further, the circumferential coordinate θ is eliminated by means of separation of variables. Thus, for instance, the θ-dependence of the normal displacement is eliminated by the assumption $w(s,\theta) = w_N \cos N\theta$ or $w(s,\theta) = w_N \sin N\theta$, as appropriate, where N is the number of circumferential waves or circumferential harmonics. The BOSOR4 code allows all required values of N to be treated in one execution of the program or in a series of separate calculations.

2. Structural Models

In most applications of ELSHOK, the modeling of the shell S for use with BOSOR4 proceeds in the straightforward manner described in Ref. [7]. Thus, the user identifies each segment of the model, includes the appropriate ring stiffeners, provides proper boundary conditions and connectivity information, specifies physical properties, and so forth. The resulting model, referred to as the "full model" in what follows, is then utilized in the modal analysis of S for all of the circumferential harmonics N included in the analysis. In some cases, however, the user may choose to highlight a portion of the shell S in order to reduce the size of the computational effort at the expense of some details of deformation beyond the region of interest. This type of analysis may also be performed using ELSHOK.

When a localized loading is applied to a portion of the shell enclosed by stiff bulkheads or isolated by heavy rings, the loading is primarily going to cause deformation near the point of impingement of the load on the shell. Any internal structure in the loaded region bounded by the stiff bulkheads or heavy rings will also be affected strongly. This pattern of deformation allows simplifications to be made regarding the portion of the structure studied carefully. Specifically, the isolated region under localized loading may be selected as the region of interest within the full model. This is accomplished in ELSHOK by employing two different mathematical models of the shell S when using BOSOR4: the full model described above and the region of interest, called the "compartment model". A description of the use of these models follows.

Modes having a circumferential wave number N=1 correspond to rigid body displacements and whipping (beam bending) of the entire structure. In order to permit the development of these important motions, the N=1 shell modes are determined using the full model. The N=0 rolling and torsional modes, those modes involving only circumferential motion of the structure, are also determined using the full model.

Deformations or motions beyond the bulkheads or rings isolating the region of interest should be unimportant for the N=0 axisymmetrical or breathing modes (normal and meridional displacements only) and for the modes corresponding to $N \geq 2$ (normal, meridional, and circumferential displacements). This is so because each bounding bulkhead or heavy ring is very stiff in its own plane and moves primarily in that plane for localized loading. Thus, these modes are determined using the compartment model, allowing unrestrained longitudinal displacements, when a simplified analysis is being performed.

In summary, the ELSHOK code allows a user to perform the modal analysis of the main body S by employing BOSOR4 in conjunction with either of two different modeling techniques. The first is the usual straightforward manner in which a mathematical model is developed and is employed for all values of N included in a given analysis. The second is an approximate technique, allowing a user to highlight a portion of the body S. This approach uses (1) the full model for the N=1 modes and the N=0 rolling and torsional modes and (2) the compartment model for the N=0 axisymmetrical or breathing modes and the modes having $N \geq 2$. Once the mathematical model of the shell S has been decided upon, the user determines the in-vacuo free-free modes and natural frequencies routinely using the BOSOR4 code.

3. Rules of Operation

For compatibility with the current version of the ELSHOK code, the following rules must be observed when executing the BOSOR4 code:

(1) Symmetry about the longitudinal midpoint of the shell may not be employed in the mathematical model of the shell unless the entire structure under consideration (including the shell S, any internal appendages σ, and any concentrated masses) is symmetrical about the midpoint.

(2) When symmetry is employed, the left-hand portion of the shell (as depicted in Fig. 1) must be modeled.

(3) When the approximate modeling technique is employed, the compartment model must be identical to the corresponding portion of the shell in the full model.

(4) The wet segments of the mathematical model (i.e., those segments representing the portion of the submerged structure in contact with the fluid) must be numbered first, starting with unity at the initial segment, at the left-hand end of the model. Any internal segments may be numbered in any manner consistent with the restrictions of Ref. [7].

(5) A separate execution of BOSOR4 must be made for each circumferential harmonic N included in an analysis. If only a full model representation of S is employed, the N=0 breathing modes and the N=0 rolling and torsional modes may be determined in the same execution of BOSOR4, if desired.

(6) Only non-negative values of N (0, +1, +2, ...) must be used. The resulting modes produced by BOSOR4 are used, with appropriate sign changes, in the subsequent shock response calculation as modes having circumferential symmetry with respect to the origin of the circumferential coordinate angle θ of Fig. 1. When antisymmetrical modes are required, they are also accounted for by means of appropriate sign changes. The sign convention used in ELSHOK, to which the components of the modes obtained from BOSOR4 are made to conform, is shown in Fig. 1.

(7) In any given shock response problem, modes must be computed and saved for at least N=0 and N=1.

4. Modifications to BOSOR4

A number of minor changes have been made to BOSOR4 to facilitate its use in the ELSHOK suite of computer programs. A brief description of each of these modifications follows:

(1) The capacity of BOSOR4 has been increased from 50 modes to 100 modes for each circumferential wave number N considered.

(2) An additional file has been supplied and assigned to logical unit 3. This file is referred to by the local or logical file name TAPE3 and corresponds to the shell mode file for a given N described in Section III. File TAPE3 contains shell geometry, physical properties, generalized masses, natural frequencies, and mode shapes, in coded format, for a given circumferential harmonic N. The shell mode file must be saved after each execution of BOSOR4 to provide data needed subsequently.

(3) The plotting routines supplied with the standard BOSOR4, and all subroutine calls involving them, have been eliminated. Plots of the mode shapes obtained from BOSOR4 are produced from the data on TAPE3 by means of a small plotting processor (BOSRMP) supplied with ELSHOK. This processor prepares plots on TEKTRONIX graphics terminals.

(4) One additional input item has been added after the usual input data. The purpose of this additional data is to provide a clean or errror-free termination of a BOSOR4 execution. Appropriate documentation may be found in the next subsection.

The above changes have been made through the use of the UPDATE batch editing system (Ref. [9]) and may be easily removed to return BOSOR4 to its standard configuration.

5. Additional Input

When using BOSOR4 with the ELSHOK code, all of the input data described in Ref. [7] must be prepared for the problem of interest. Moreover, one additional input item should be placed after the usual set of data to terminate the execution of BOSOR4 without error. A description of this input item follows:

ITEM NO.	VARIABLE	FORMAT
last	TITLE(1)	A10

where TITLE(1) = first word of the alphameric title array employed by BOSOR4. The user should supply the value ENDbDATAbb, in which the symbol "b" denotes a blank space.

Note: If TITLE(1)=ENDbDATAbb, an end of file mark is written on logical file TAPE3, and the execution of BOSOR4 is terminated by a call to the system routine EXIT. If this value is not supplied, the execution of BOSOR4 proceeds normally but terminates with a system error indicating an attempt to read more input data than supplied. The use of this additional input item is optional.

6. Sample Control Cards

A typical set of control cards or job control language (JCL) statements for an execution of BOSOR4 under a NOS 1 operating system is shown below. A brief outline of these control cards is presented in the next subsection. Various other arrangements of the job steps are possible.

```
BOSOR4(T277) RANLET
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(1777)
COPYBR(INPUT,INFILE)
REWIND(INFILE)
PURGE(BOSSY00/NA)
DEFINE(TAPE3=BOSSY00/CT=S,M=R)
ATTACH(LGO=BOSFTN)
RFL(125000)
LDSET(PRESET=INDEF,MAP=N)
LGO(INFILE)
GOTO(CATALOG)
EXIT.
CATALOG(TAPE3,R)
REWIND(TAPE3,INFILE)
COPYBF(TAPE3,COPY)
COPYSBF(INFILE,COPY)
ROUTE(COPY,DC=PR,TID=WEIDLNG)
```

7. Description of Control Cards

```
BOSOR4(T277) RANLET
```
Job card or job statement containing an alphameric job name, a time limit in seconds (octal), and an optional identifier.

```
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(1777)
```
System dependent accounting information.

```
COPYBR(INPUT,INFILE)
REWIND(INFILE)
```
Provide copy of input data card images on local file INFILE.

```
ATTACH(LGO=BOSFTN)
```
Assign permanent file BOSFTN to current job with local file name LGO. File BOSFTN contains the compiled code or object module for BOSOR4.

```
PURGE(BOSSY00/NA)
DEFINE(TAPE3=BOSSY00/CT=S,M=R)
```
Assign space on permanent file device for the shell mode file (TAPE3) under the permanent file name BOSSY00. Information written on TAPE3 during the execution of BOSOR4 is written directly on the permanent file.

```
RFL(125000)
```
Request field length for execution. This is optional on many systems.

```
LDSET(PRESET=INDEF,MAP=N)
```
Select load options.

```
LGO(INFILE)
```
Load and execute the BOSOR4 object module. The required set of input data is supplied on local file INFILE.

```
GOTO(CATALOG)
```
Unconditional transfer to control statement CATALOG if the execution terminates normally.

```
EXIT.
```
Provide transfer of control when an execution error occurs. If this statement is encountered after an error-free program execution, the system terminates the job. If an error condition occurs during job execution, the system processes the control stat ments that follow the EXIT statement.

```
CATALOG(TAPE3,R)
```
List information about each logical record of file TAPE3 on file OUTPUT.

```
REWIND(TAPE3,INFILE)
COPYBF(TAPE3,COPY)
COPYSBF(INFILE,COPY)
```
Copy the information on files TAPE3 and INFILE to the local file COPY.

```
ROUTE(COPY,DC=PR,TID=WEIDLNG)
```
Dispose the file COPY to the remote site WEIDLNG. At job termination, the system changes the local file OUTPUT into a print file which is disposed to the site corresponding to the job origin.

8. Input Data for Sample Problem 1

A set of input data applicable to Sample Problem 1 is shown below. These input cards or card images, depending on the mode of operation selected by the user, direct the BOSOR4 code to perform the modal analysis of the shell S of Fig. 4 for the circumferential harmonic N=0. For each of the other circumferential harmonics employed (N=1,2,3), the parameters NMINB and NMAXB on the fifth card must be set to the value of N being considered and the alphameric job title must be modified accordingly.

```
RING-STIFFENED CYLINDER WITH FLAT END CAPS -- SYMMETRY USED -- N = 0
     2     2    -1     0     0                    INDIC,NPRT,NLAST,ISTRES,IPRE
     2     3     0     0                          NSEG,NCOND,IBOUND,IRIGID
     0     0     0                                NSTART,NFIN,INCR
     0     0     0     1    30                    NDB,NMINB,NMAXB,INCRB,NVEC
     0     0     0                                NDIST,NCIRC,NTHETA
     0                                            (ITHETA(I), I = 1,NCIRC)
          0.0                                     (THETA(I), I = 1,NDIST)
          0.0          0.0          0.0           THETAM,THETAS,PREROT
001001001001     0     0     0     0        0.0          0.0              IFIX
001010002001     1     1     1     1        0.0          0.0              IFIX
002051002051     1     0     0     1        0.0          0.0              IFIX
          0.0          0.0          0.0        0.0          P,STEP,TEMP,DTEMP
          0.0          0.0          0.0                     FSTART,FMAX,DF
    10     3     0                    SEGMENT  1          NMESH,NTYPEH,INTVAL
     1     3     0                                        NSHAPE,NTYPEZ,IMP
          0.0          0.0     16.8125        0.0          R1,Z1,R2,Z2
         2.50                                             ZVAL
     0                                                    NRINGS
     0                                                    LINTYP
     0     0     0     0                    NLTYPE,NPSTAT,NTSTAT,NTGRAD
     2                                                    NWALL
    10.8E 06          0.32   2.535E-04        0.0          0.0          1.0E,U,SM,A
    51     3     0                    SEGMENT 2          NMESH,NTYPEH,INTVAL
     1     3     0                                        NSHAPE,NTYPEZ,IMP
     16.8125          0.0     16.8125    127.8125          R1,Z1,R2,Z2
      0.1875                                              ZVAL
     2                                                    NRINGS
     2                                                    NTYPE
      51.875       102.50                                       Z(I),I=1,NRINGS
     1     1                                                 NTYPER(I),I=1,NRINGS
    30.0E 06          8.25      11.698        2.75          0.0       -2.0625
          0.0     8.72E 07   7.33E-04                    E,A,IY,IX,IXY,E1,E2,GJ,RM
    30.0E 06          8.25      11.698        2.75          0.0       -2.0625
          0.0     8.72E 07   7.33E-04                    E,A,IY,IX,IXY,E1,E2,GJ,RM
     0                                                    LINTYP
     0     0     0     0                    NLTYPE,NPSTAT,NTSTAT,NTGRAD
     2                                                    NWALL
    30.0E 06          0.30   7.33E-04          0.0          1.0          1.0E,U,SM,A
     1     1     0     0                    IRECT1,IRECT2,IVAR1,IVAR2
     0     0                                N1,K1
    30.0E 06          0.30   7.33E-04                    E1,U1,STIFMD
          0.0          0.0                              T1,M1
     0                                                    K2
    30.0E 06          0.30   7.33E-04                    E2,U2,RGMD
        5.625          0.25       1.25                  D2,T2,M2
END DATA
```

9. Input Data for Sample Problem 2

The set of input data listed below directs the BOSOR4 code to perform the modal analysis of the full model of the shell S, shown schematically in Fig. 14, for the circumferential harmonic N=1.

```
SYMMETRIC RING-STIFFENED SHELL WITH FLAT ENDS (FULL MODEL) -- N = 1
     2      2     -1      0      0                          INDIC,NPRT,NLAST,ISTRES,IPRE
     3      4      0      0                                 NSEG,NCOND,IBOUND,IRIGID
     0      0      0                                        NSTART,NFIN,INCR
     0      1      1      1     30                          NOB,NMINB,NMAXB,INCRB,NVEC
     0      0      0                                        NDIST,NCIRC,NTHETA
     0                                                     (ITHETA(I), I = 1,NCIRC)
          0.0                                              (THETA(I), I = 1,NDIST)
          0.0          0.0          0.0                    THETAM,THETAS,PREROT
001001001001      0      0      0      0          0.0            0.0              IFIX
001010002001      1      1      1      1          0.0            0.0              IFIX
002035003001      1      1      1      1          0.0            0.0              IFIX
003017003017      1      0      0      1          0.0            0.0              IFIX
          0.0          0.0          0.0          0.0              P,STEP,TEMP,DTEMP
          0.0          0.0          0.0                           FSTART,FMAX,DF
    10      3      0                          SEGMENT  1          NMESH,NTYPEH,INTVAL
     1      3      0                                             NSHAPE,NTYPEZ,IMP
          0.0          0.0      16.8125          0.0             R1,Z1,R2,Z2
         2.50                                                    ZVAL
     0                                                           NRINGS
     0                                                           LINTYP
     0      0      0      0                                 NLTYPE,NPSTAT,NTSTAT,NTGRAD
     2                                                           NWALL
    10.8E 06         0.32    2.535E-04          0.0            0.0          1.0E,U,SM,A
    35      3      0                          SEGMENT  2          NMESH,NTYPEH,INTVAL
     1      3      0                                             NSHAPE,NTYPEZ,IMP
    16.8125          0.0      16.8125       102.50              R1,Z1,R2,Z2
     0.1875                                                      ZVAL
     1                                                          NRINGS
     2                                                          NTYPE
     51.875                                                           Z(I),I=1,NRINGS
     1                                                          NTYPER(I),I=1,NRINGS
    30.0E 06         8.25      11.698          2.75            0.0        -2.0625
          0.0     8.72E 07    7.33E-04                      E,A,IY,IX,IXY,E1,E2,GJ,RM
     0                                                          LINTYP
     0      0      0      0                                 NLTYPE,NPSTAT,NTSTAT,NTGRAD
     2                                                          NWALL
    30.0E 06         0.30    7.33E-04          0.0            1.0          1.0E,U,SM,A
     1      1      0      0                                 IRECT1,IRECT2,IVAR1,IVAR2
     0      0                                               N1,K1
    30.0E 06         0.30    7.33E-04                       E1,U1,SFIFMD
          0.0          0.0                                  T1,M1
     0                                                      K2
    30.0E 06         0.30    7.33E-04                       E2,U2,RGMD
     5.625           0.25        1.25                       D2,T2,M2
    17      3      0                          SEGMENT  3          NMESH,NTYPEH,INTVAL
     1      3      0                                             NSHAPE,NTYPEZ,IMP
    16.8125       102.50      16.8125       127.8125            R1,Z1,R2,Z2
     0.1875                                                      ZVAL
     1                                                          NRINGS
     4                                                          NTYPE
          0.0                                                         S(I),I=1,NRINGS
     1                                                          NTYPER(I),I=1,NRINGS
    30.0E 06         8.25      11.698          2.75            0.0        -2.0625
          0.0     8.72E 07    7.33E-04                      E,A,IY,IX,IXY,E1,E2,GJ,RM
```

(CONTINUED ON NEXT PAGE)

```
      0                                                    LINTYP
      0      0      0      0                          NLTYPE,NPSTAT,NTSTAT,NTORAD
      2                                                   NWALL
     30.0E 06          0.30     7.33E-04       0.0          1.0          1.0E,U,SM,A
      1      1      0      0                          IRECT1,IRECT2,IVAR1,IVAR2
      0      0                                         N1,K1
     30.0E 06          0.30     7.33E-04               E1,U1,STIFMD
          0.0           0.0                           T1,H1
      0                                                K2
     30.0E 06          0.30     7.33E-04               E2,U2,RGMD
        5.625          0.25         1.25               D2,T2,H2
END DATA
```

For the circumferential harmonic N=0, the set of input data shown below directs the BOSOR4 code to perform the modal analysis of the compartment model of the shell S (Fig. 14). For N=2 and N=3, the parameters NMINB and NMAXB on the fifth card must be set to the value of N being considered and the alphameric job title must be modified accordingly.

```
SYMMETRIC RING-STIFFENED SHELL WITH FLAT ENDS (CMPT MODEL) -- N = 0
      2      2     -1      0      0                    INDIC,NPRT,NLAST,ISTRES,IPRE
      1      2      0      0                          NSEG,NCOND,IBOUND,IRIGID
      0      0      0                                 NSTART,NFIN,INCR
      0      0      0      1     20                   NOB,NMINB,NMAXB,INCRB,NVEC
      0      0      0                                 NDIST,NCIRC,NTHETA
      0                                               (ITHETA(I), I = 1,NCIRC)
           0.0                                        (THETA(I), I = 1,NDIST)
           0.0           0.0           0.0            THETAM,THETAS,PREROT
001001001001      0      0      0      0       0.0          0.0            IFIX
001017001017      1      0      0      1       0.0          0.0            IFIX
           0.0           0.0           0.0       0.0        P,STEP,TEMP,DTEMP
           0.0           0.0           0.0                  FSTART,FMAX,DF
     17      3      0                        SEGMENT  1     NMESH,NTYPEH,INTVAL
      1      3      0                                      NSHAPE,NTYPEZ,IMP
     16.8125       102.50       16.8125     127.8125       R1,Z1,R2,Z2
      0.1875                                              ZVAL
      1                                                   NRINGS
      4                                                   NTYPE
           0.0                                                  S(I),I=1,NRINGS
      1                                                    NTYPER(I),I=1,NRINGS
     30.0E 06          8.25        11.698       2.75          0.0         -2.0625
           0.0      8.72E 07     7.33E-04                E,A,IY,IX,IXY,E1,E2,GJ,RM
      0                                                   LINTYP
      0      0      0      0                          NLTYPE,NPSTAT,NTSTAT,NTORAD
      2                                                   NWALL
     30.0E 06          0.30     7.33E-04       0.0          1.0          1.0E,U,SM,A
      1      1      0      0                          IRECT1,IRECT2,IVAR1,IVAR2
      0      0                                         N1,K1
     30.0E 06          0.30     7.33E-04               E1,U1,STIFMD
          0.0           0.0                           T1,H1
      0                                                K2
     30.0E 06          0.30     7.33E-04               E2,U2,RGMD
        5.625          0.25         1.25               D2,T2,H2
END DATA
```

APPENDIX B - THE USE OF ACESNID

1. Background

The ACESNID computer code of Ref. [10] is included in the ELSHOK suite of computer programs for determining the virtual mass array, a quantity producing the late-time contribution of the DAA. Spatial functions having the property of orthogonality over the wet surface of a submerged body are required in the formulation upon which ACESNID is based. These surface expansion functions are obtained during the calculation of the virtual or entrained mass. The ACESNID code is quite general and considers a cavity in an infinite acoustic fluid corresponding to the wet surface of a fully submerged shell of revolution having an arbitrary meridional geometry. A knowledge of the information given in Refs. [2] and [10] and Sections II and III of the present report will aid in the understanding of this appendix.

In ACESNID, the motion of the fluid induced by prescribed steady-state vibrations is governed by a velocity potential which satisfies the Helmholtz equation. The interior boundary of the fluid, a cavity of revolution, is divided into bands. A line distribution of simple sources is placed on the central line of each band, and acoustic influence coefficients for pressure and velocity are determined for a given frequency of excitation. The source strengths are then evaluated by equating the normal fluid velocity on the surface of the cavity to prescribed values corresponding to steady-state motions with spatial amplitude distributions given by surface expansion functions. From the source strengths follow the fluid pressures developed on the surface

of the cavity and the generalized forces required to determine the accession to inertia and damping associated with the chosen surface expansion functions.

When applying ACESNID with ELSHOK, a steady-state problem of sufficiently low frequency must be considered to approximate the incompressible behavior of the fluid. In such a case, the solution to the Helmholtz equation approaches the solution to the Laplace equation. The resulting accession to damping becomes negligible, and the accession to inertia yields the virtual mass array, as discussed in Ref. [2]. The formulation employed in ACESNID is based upon a separation of variables, and allows the user to treat all of the required values of the circumferential harmonic N in one execution or in a series of separate calculations.

2. Partitioning of Cavity

In all applications of ELSHOK, the use of ACESNID must parallel the use of BOSOR4. If the structural modeling proceeded in the straightforward manner of using only the full model representation of the shell S, consistent partitioning of the cavity in the fluid must be provided for the generation of the surface expansion functions. Thus, the user may not highlight a region of interest. On the other hand, if the user chose to highlight a portion of the structural model and to apply the full and compartment models in the fashion described in Section II and Appendix A, the partitioning of the cavity must reflect the highlighted region of interest to allow the evaluation of the appropriate surface expansion functions.

When generating surface expansion functions for use with the straightforward full model representation of the shell (no region of interest), the cavity in the fluid, or the wet surface of the submerged

structure, must in general be partitioned into three distinct regions: (1) the left region corresponding to the left end closure; (2) the right region corresponding to the right end closure; and (3) the central region, which need not be a right circular cylinder, lying between the left and right regions. When symmetry about the longitudinal midpoint of the structure is employed in the full model, specification of the right region is not required. In the straightforward mode of operation, all circumferential harmonics considered in a problem under study may be treated in one execution of ACESNID. Equations (13)-(15) govern the generation of the surface expansion functions for this case.

When generating surface expansion functions for use with the simplified representation of the shell, the cavity in the fluid must in general be partitioned into four distinct regions. The first three are identical to those described above. The fourth region corresponds to the compartment model or region of interest within the full model of the shell S. Because different requirements must be met for N=1 and for N≠1, as discussed in Section II and Appendix A, a separate execution of ACESNID is required for each value of N included in a given analysis. Equations (13)-(16) govern the generation of the surface expansion functions for this case.

3. Rules of Operation

For compatibility with the current version of the ELSHOK code, the following rules must be observed when executing the ACESNID code:

(1) The use of the BOSOR4 code must precede the use of ACESNID. This is so because the ACESNID code obtains the geometry of the cavity in the fluid from a shell mode file for a full model.

(2) The values of the circumferential harmonic N employed with ACESNID must be identical to those used with BOSOR4. If N=0, 1,2,3 were used for the modal structural analysis of the main body S, then N=0,1,2,3 must be used for the calculation of the virtual mass.

(3) For the generation of the surface expansion functions, the partitioning of the cavity in the fluid must be described with reference to the full model of the shell S.

(4) The frequency of excitation specified by the user must be sufficiently small that the behavior of the fluid in the ACESNID calculation approximates that of an incompressible fluid.

(5) When a compartment region or region of interest is considered, a separate execution of ACESNID must be performed for each circumferential wave number N included in an analysis. When straightforward modeling, involving only a full model, is employed, all required values of N may be treated in one execution of ACESNID.

(6) The virtual mass file must contain the results computed for each circumferential harmonic N in the order of increasing number of circumferential waves (i.e., in the order N=0, N=1, N=2,...).

4. Modifications to ACESNID

Reference [10] documents a version of ACESNID prepared for use with the EPSA code of Ref. [1]. The version of ACESNID supplied in the ELSHOK suite of computer programs differs somewhat from the version of Ref. [1]. A brief description of each of the major differences between these two versions of the ACESNID code follows:

(1) Any finite cavity of revolution may be considered with ELSHOK.

(2) The input data specifications have been changed. Increased flexibility has been provided to allow description of more general cavities of revolution for both the generation of the surface

expansion functions and the calculation of the acoustic influence coefficients. The source regions, which pertain to the calculation of the acoustic influence coefficients, are described or partitioned separately from the regions associated with the surface expansion functions.

(3) Numerical integrations are performed using the 1/3 rule due to Simpson (Ref. [12]). The trapezoidal integration rule employed in Ref. [10] has been replaced. Some reduction in execution time has been observed as a result of this change.

(4) A file has been supplied and assigned to logical unit 4. This file is referred to by the local or logical-file name TAPE4. File TAPE4 is a shell mode file (TAPE3 from a BOSOR4 calculation) for a full model of the shell S. It is used in ACESNID to describe the geometry of the cavity in the fluid.

(5) File TAPE7 has been added and assigned to logical unit 7. This file receives the virtual mass array in coded card-image format for each circumferential harmonic N considered in an execution of ACESNID. File TAPE7 must be saved to provide data needed subsequently.

5. Description of Input Data

A description of the input data required for an execution of the ACESNID code is given below. The input data specifications are written in a style similar to FORTRAN. It is assumed that the user is familiar with this programming language.

Input Set 1 — FORMAT (8A10)

TITLE

where TITLE = alphameric job title or job heading.

Input Set 2 — FORMAT (5I5)

NSTART, NFINIS, NFREQ, NVMASS, NCHECK

where NSTART = initial circumferential harmonic N to be used in the calculation.
NFINIS = final circumferential harmonic N to be used in the calculation. Calculations are done for an increment of unity in the circumferential harmonic.
NFREQ = number of excitation frequencies. For ELSHOK applications, use NFREQ = 1.
NVMASS = integer flag indicating whether or not a calculation of virtual mass is to be performed (1=yes; 0=no). For ELSHOK applications, use NVMASS = 1.
NCHECK = integer flag indicating whether or not the computed source strengths are to be checked by back substitution into the linear algebraic equations governing their solution (1=yes; 0=no). For most ELSHOK applications, use NCHECK = 0.

NOTE: Since ACESNID utilizes main memory management, it is quite easy to reduce the core requirement or central memory to the minimum size required for a given execution. This may be accomplished by specifying a negative value of NSTART in an otherwise complete set of input data. An execution of ACESNID with this set of input data will then supply information from which the desired size of the dynamic array may be determined. When determining the minimum allowable memory size for a given problem, the dimension of the dynamic array in blank common must be set to at least NSEG+NFREQ+1 words, where NSEG is the number of segments in the shell mode file supplied as TAPE4.

INPUT SET 3 FORMAT (5I5)

NLEFT, NCYL, NRITE, NWBOSG, NSYMF

where NLEFT = number of bands on the left end closure of the cavity in the fluid.

NCYL = number of bands on the cylindrical portion of the cavity in the fluid. If no cylindrical portion exists for the problem under study, use NCYL = 0.

NRITE = number of bands on the right end closure of the cavity in the fluid. This value must be supplied, since ACESNID requires the entire cavity in the fluid, even if a model of half of the submerged shell S is provided on TAPE4 because of symmetry.

NWBOSG = number of wet segments in the BOSOR4 model contained on the shell mode file supplied as TAPE4.

NSYMF = integer flag indicating whether or not the cavity in the fluid must be completed using symmetry (1=yes; 0=no). If the geometry of the full model on TAPE4 describes half of an entire structural model, then use NSYMF = 1.

Note 1 : The input parameters NLEFT, NCYL, NRITE are used to partition the entire interior boundary of the fluid into, in general, three distinct regions which may be termed "source regions". The cylindrical portion of the cavity in the fluid may not be present in a given problem. In essence, these input parameters specify the number of lines of simple sources employed to describe the behavior of the fluid. The use of source regions allows the acoustic influence coefficients to be computed in an efficient manner.

Note 2 : Although the source band parameters do not apply to the generation of surface expansion functions, their values are governed by the number of surface expansion functions selected. This is so because compatibility of normal motion is required at the structure-fluid interface. Thus, a sufficient number of source bands must be provided to enable the fluid response to match the normal motion specified by the surface expansion functions. For most applications of ACESNID, about four or five bands are needed in the shortest half-wavelength or shortest local undulation found in the set of surface expansion functions.

Note 3: In many applications of ACESNID, the end closures of the fluid cavity correspond to the end closures of the structural model, and the central portion of the shell of revolution is a circular cylinder. However, this is not a requirement. The user is advised to refer to Ref. [10] for more detailed information.

Input Set 4 FORMAT (5I5)

NORDER, NFENDS, NFCENT, NFCMPT, NOMIT

where NORDER = control integer for generation of surface expansion functions.
NORDER = 1 : surface expansion functions are alternately symmetrical and antisymmetrical about the midpoint of the structure. Use this option for a general unsymmetrical problem.
NORDER = 2 : surface expansion functions are symmetrical about the midpoint of the structure. Use this option for a symmetrical problem.
NORDER = 3 : surface expansion functions are antisymmetrical about the midpoint of the structure. Use this option for an antisymmetrical problem.

NFENDS = number of surface expansion functions to be generated on the end closures of the structure. An equal number of surface expansions is generated on each end closure unless indicated otherwise by input parameter NOMIT.

NFCENT = number of surface expansion functions to be generated on the central portion of the structure.

NFCMPT = number of surface expansion functions to be generated on the compartment portion of the structure. If a compartment model is not employed, used NFCMPT = 0.

NOMIT = control parameter for omitting surface expansion functions on end closures of structure.
NOMIT = 0 : no surface expansion functions omitted.
NOMIT = 1 : no surface expansion functions generated on left end closure of structure.
NOMIT = 2 : no surface expansion functions generated on right end closure of structure.

Input Set 5 FORMAT (3E10.0)

RHOFL, VSOUND, ERR

where RHOFL = mass density of fluid.

VSOUND = speed of sound in fluid.

ERR = relative accuracy accepted for numerical integration. This parameter terminates the numerical integration procedure used in ACESNID. In general, use $0.005 \leq ERR \leq 0.0075$.

Input Set 6 FORMAT (10I5)

LGEONP, LFCNNP, LSEFNP, LORTH, LGEOSP,
LSEFSP, LINF, LRHS, LSOL, LACID

where LGEONP = output flag for structural nodal point geometry.
LFCNNP = output flag for convenient functions used to generate surface expansion functions.
LSEFNP = output flag for surface expansion functions evaluated at structural nodal points.
LORTH = output flag for orthogonality check of surface expansion functions.
LGEOSP = output flag for geometry of source bands.
LSEFSP = output flag for surface expansion functions evaluated at central lines of source bands.
LINF = output flag for acoustic influence coefficients.
LRHS = output flag for right-hand sides of equations for determining source strengths.
LSOL = output flag for source strengths.
LACID = output flag for accession to inertia and damping.

Note 1 : In an output flag is set to 1, output is supplied. If an output flag is set to 0, no output is supplied.

Note 2 : In almost all cases, use LINF = 0, since the acoustic influence coefficients are very large complex arrays for most problems of interest.

Input Set 7 FORMAT (8E10.0)

(CPS(J), J = 1, NFREQ)

where CPS(J) = the J-th frequency of excitation. For ELSHOK applications, supply only CPS(1), since NFREQ = 1, as stipulated above.

Note: The frequency of excitation supplied as CPS(1) must be sufficiently low to produce steady-state incompressible behavior of the fluid. For ACESNID, it has been found that an excitation frequency is sufficiently low when the quantity $\omega_E a_M/c$ is of the order of 0.0001 or smaller. Here, ω_E is the circular frequency of excitation, a_M is the maximum radius of the central portion of the shell S, and c is the speed of sound in the fluid.

Input Set 8 FORMAT (4I5)

```
   DO 10 K = 1, 4
10 READ  JSGBEG, JPTBEG, JSGEND, JPTEND
```

where JSGBEG = segment number defining the exordium of the K-th region for generating surface expansion functions.

JPTBEG = nodal point, of segment JSGBEG, defining the exordium of the K-th region for generating surface expansion functions.

JSGEND = segment number defining the terminus of the K-th region for generating surface expansion functions.

JPTEND = nodal point, of segment JSGEND, defining the terminus of the K-th region for generating surface expansion functions.

K = 1 : supply segment and nodal point numbers for wet surface of left end closure of structure S.

K = 2 : supply segment and nodal point numbers for wet surface of central region of structure S.

K = 3 : supply segment and nodal point numbers for wet surface of right end closure of structure S.

K = 4 : supply segment and nodal point numbers for wet surface of compartment region of structure S.

Note 1 : The segment and nodal point numbers mentioned above refer to the numeration supplied by the BOSOR4 code on a shell mode file for a full model representation of the main body S. The nodal points are actually points on the reference meridian used to generate the shell of revolution S. They are numbered, starting from unity, within each segment.

Note 2 : If the number of surface expansion functions associated with a given K-th region above is zero, omit the input parameters for that value of K. If NSYMF = 1, omit the input parameters for the right end closure (K = 3).

Input Set 9 FORMAT (4I5)

```
    DO 20 K = 1, 3
20  READ  JSGBEG, JPTBEG, JSGEND, JPTEND
```

where JSGBEG = segment number defining the exordium of the K-th source region for evaluating acoustic influence coefficients.

JPTBEG = nodal point, of segment JSGBEG, defining the exordium of the K-th source region for evaluating acoustic influence coefficients.

JSGEND = segment number defining the terminus of the K-th source region for evaulating acoustic influence coefficients.

JPTEND = nodal point, of segment JSGEND, defining the terminus of the K-th source region for evaluating acoustic influence coefficients.

K = 1 : supply segment and nodal point numbers for left end closure of cavity in fluid.

K = 2 : supply segment and nodal point numbers for cylindrical portion of cavity in fluid.

K = 3 : supply segment and nodal point numbers for right end closure of cavity in fluid.

Note 1 : The segment and nodal point numbers mentioned above refer to the numeration supplied by the BOSOR4 code on a shell mode file for a full model representation of the main body S. The nodal points are actually points on the reference meridan used to generate the shell of revolution S. They are numbered, starting from unity, within each segment.

Note 2 : If NSYMF = 1, omit the input parameters for the right end closure (K = 3).

6. Sample Control Cards

A sample set of control cards or job control language statements for an execution of ACESNID under a NOS 1 operating system is shown below. The next subsection contains a brief description of these control statements. Various other arrangements of the job steps are possible.

```
ACESNID(T2777) RANLET
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(7777)
ATTACH(OLDPL=ACESNID)
UPDATE(F,L=A124)
RETURN(OLDPL)
FTN(I=COMPILE,L=0,OPT=2,A)
RETURN(COMPILE)
ATTACH(TAPE4=BOSSY00)
RFL(165000)
LDSET(PRESET=INDEF,MAP=SB)
LGO.
GOTO(PURGE)
EXIT.
PURGE(VIRMAS/NA)
SAVE(TAPE7=VIRMAS/CT=S,M=R)
REWIND(TAPE7)
COPYSBF(TAPE7,OUTPUT)
```

If control statements of the above type are used to perform a virtual mass calculation for a given circumferential harmonic N, the files saved from the local file TAPE7 for the individual values of N may be merged to produce the virtual mass file required by ELSHOK in the following manner:

```
VMFILE(T7) RANLET
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
GET(VIRMAS0,VIRMAS1,VIRMAS2,VIRMAS3)
COPYBR(VIRMAS0,VMFILE)
COPYBR(VIRMAS1,VMFILE)
COPYBR(VIRMAS2,VMFILE)
COPYBR(VIRMAS3,VMFILE)
PURGE(VIRMAS/NA)
SAVE(VMFILE=VIRMAS/CT=S,M=R)
EXIT.
```

7. Description of Control Cards

```
ACESNID(T2777) RANLET
```
Job card or job statement.
```
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(7777)
```
System dependent accounting information.
```
ATTACH(OLDPL=ACESNID)
```
Assign permanent file ACESNID to current job with local file name OLDPL. File ACESNID contains the UPDATE program library for the ACESNID code.
```
UPDATE(F,L=A124)
RETURN(OLDPL)
```
Process UPDATE directives supplied in the job stream (on file INPUT) and prepare a source file suitable for compilation.
```
FTN(I=COMPILE,L=0,OPT=2,A)
RETURN(COMPILE)
```
Produce compiled code or object module for the ACESNID code.
```
ATTACH(TAPE4=BOSSY00)
```
Assign permanent file BOSSY00 to current job with local file name TAPE4. File BOSSY00 contains a shell mode file for the full model representation of the shell S.
```
RFL(165000)
```
Request field length for execution. This is optional on many systems.
```
LDSET(PRESET=INDEF,MAP=SB)
LGO.
```
Select load options, load, and execute the compiled version of ACESNID. The required set of input data is supplied in the job stream (on file INPUT).
```
GOTO(PURGE)
```
Unconditional transfer to control statement PURGE if the execution terminates normally.
```
EXIT.
```
Terminate execution or provide transfer of control when an execution error occurs.
```
PURGE(VIRMAS/NA)
SAVE(TAPE7=VIRMAS/CT=S,M=R)
```
Save local file TAPE7 on permanent file device under the permanent file name VIRMAS. File TAPE7 contains the virtual mass information from the current execution of ACESNID.
```
REWIND(TAPE7)
COPYSBF(TAPE7,OUTPUT)
```
Provide copy of local file TAPE7 on the OUTPUT file.

Note: The job control statements for merging the results of several ACESNID calculations into a single file (virtual mass file) are self-explanatory.

8. UPDATE Directives and Input Data for Sample Problem 1

The UPDATE directives employed to calculate the virtual mass for Sample Problem 1 are listed below.

```
*IDENT,STORAGE
*DELETE,ACESNID.26,ACESNID.27
      COMMON A(40000)
      NWORDA = 40000
```

The size of the dynamic array A shown above is slightly in excess of the size determined by a restricted execution of the ACESNID code. In this restricted execution, the dimension of the array A and the variable NWORDA were set to 50 words and the input parameter NSTART was set to -1. All other input parameters were identical to those listed below. When the dynamic array is dimensioned to 40000 words, as above, a field length of almost 165000 words (octal) is required for an execution of the ACESNID code.

The set of input data shown below directs the ACESNID code to perform the virtual mass calculation for Sample Problem 1 for the circumferential harmonics N=0,1,2,3. All references to shell geometry correspond to the structural data presented in Appendix A for Sample Problem 1.

```
VIRTUAL MASS FOR RING-STIFFENED CYLINDER WITH FLAT END CAPS -- N = 0,1,2,3
    0    3    1    1    0     NSTART,NFINIS,NFREQ,NVMASS,NCHECK
   12   75   12    2    1     NLEFT,NCYL,NRITE,NWBOSG,NSYMF
    2    3   15    0    0     NORDER,NFENDS,NFCENT,NFCMPT,NOMIT
 9.45216-5    60.0+3     0.005          RHOFL,VSOUND,ERR
    1    1    1    1    0     0    0     0    1    1     OUTPUT FLAGS
      0.10                   CPS(1)
    1    1    1   12    SEFS       JSGBEG,JPTBEG,JSGEND,JPTEND    LEFT
    2    1    2   53    SEFS       JSGBEG,JPTBEG,JSGEND,JPTEND    CENTRAL
    1    1    1   12    SOURCES    JSGBEG,JPTBEG,JSGEND,JPTEND    LEFT
    2    1    2   53    SOURCES    JSGBEG,JPTBEG,JSGEND,JPTEND    CYLINDER
```

9. UPDATE Directives and Input Data for Sample Problem 2

The UPDATE directives employed to calculate the virtual mass for Sample Problem 2 are listed below.

```
*IDENT,STORAGE
*DELETE,ACESNID.26,ACESNID.27
      COMMON A(40000)
      NWORDA = 40000
```

The size of the dynamic array A shown above is slightly in excess of the size determined by a restricted execution of the ACESNID code. In this restricted execution, the dimension of the array A and the variable NWORDA were set to 50 words and the input parameter NSTART was set to -1. All other input parameters were identical to those listed below. When the dynamic array is dimensioned to 40000 words, as above, a field length of almost 165000 words (octal) is required for an execution of the ACESNID code.

The set of input data shown below directs the ACESNID code to perform the virtual mass calculation for Sample Problem 2 for the circumferential harmonic N=1. The virtual mass so produced is intended for use with the full model representation of the shell S. All references to shell geometry correspond to the full structural model presented in Appendix A for Sample Problem 2.

```
VIRTUAL MASS FOR CYLINDER WITH FLAT ENDS (FULL MODEL) -- N = 1
    1    1    1    1    0     NSTART,NFINIS,NFREQ,NVMASS,NCHECK
   10   75   10    3    1     NLEFT,NCYL,NRITE,NWBOSS,NSYMF
    2    3    8    7    0     NORDER,NFENDS,NFCENT,NFCMPT,NOMIT
9.45216-5    60.0+3    0.0075          RHOFL,VSOUND,ERR
    1    1    1    1    0    0    0    0    1    1    OUTPUT FLAGS
  0.10                       CPS(1)
    1    1    1   12    SEFS       JSOBEG,JPTBEG,JSOEND,JPTEND   LEFT
    2    1    3   19    SEFS       JSOBEG,JPTBEG,JSOEND,JPTEND   CENTRAL
    3    1    3   19    SEFS       JSOBEG,JPTBEG,JSOEND,JPTEND   CMPT
    1    1    1   12    SOURCES    JSOBEG,JPTBEG,JSOEND,JPTEND   LEFT
    2    1    3   19    SOURCES    JSOBEG,JPTBEG,JSOEND,JPTEND   CYLINDER
```

For the circumferential harmonic N=0, the virtual mass associated with the compartment model of the shell S may be determined by executing the ACESNID code with the set of input data listed below. For N=2 and N=3, which must be treated separately for Sample Problem 2, the parameters NSTART and NFINIS must be set to the value of N being considered and the alphameric job title must be modified accordingly.

```
VIRTUAL MASS FOR CYLINDER WITH FLAT ENDS (CMPT MODEL) -- N = 0
    0     0     1     1     0     NSTART,NFINIS,NFREQ,NVMASS,NCHECK
   10    75    10     3     1     NLEFT,NCYL,NRITE,NWBOSG,NSYMF
    2     0     0     7     0     NORDER,NFENDS,NFCENT,NFCMPT,NOMIT
9.45216-5     60.0+3     0.0075          RHOFL,VSOUND,ERR
    1     1     1     1     0     0     0     0     1     1     OUTPUT FLAGS
      0.10                      CPS(1)
    3     1     3    19     SEFS        JSGBEG,JPTBEG,JSGEND,JPTEND    CMPT
    1     1     1    12     SOURCES     JSGBEG,JPTBEG,JSGEND,JPTEND    LEFT
    2     1     3    19     SOURCES     JSGBEG,JPTBEG,JSGEND,JPTEND    CYLINDER
```

APPENDIX C - THE USE OF PIFLASH

1. General Information

The PIFLASH computer code is a post-processor for the BOSOR4 and ACESNID codes. As discussed in Section III, the PIFLASH code takes data from all of the shell mode files (produced by BOSOR4) and from the virtual mass file (produced by ACESNID) and reorganizes it to facilitate the solution of the equations governing the transient response problem under investigation.*) Any concentrated or point masses which are attached to the shell S are accounted for during the execution of PIFLASH. Familiarity with Appendices A and B will aid in the understanding of this appendix.

The labels or headings employed to identify the various portions of the output supplied by PIFLASH correspond to the terminology used with the approximate modeling technique available for the shell S, even if the straightforward modeling technique (with no region of interest) is being used in a particular problem. Thus, geometry, mode shapes, surface expansion functions, etc are labeled as applying to the full model when they are associated with files produced for N=0 (torsional) and N=1. Items associated with N=0 (breathing) and $N \geq 2$ are labeled as applying to the compartment model.

---

*) The current version of ELSHOK constructs the equations governing the shock response problem for the entire structure and for the entire cavity in the fluid, even if symmetry is employed in a given analysis. Future versions of ELSHOK may be based upon a different approach.

2. Rules of Operation

(1) Data from the BOSOR4 and ACESNID codes must be available prior to the execution of the PIFLASH code.

(2) Data must be supplied to PIFLASH for at least N=0 (breathing) and N=1.

(3) The virtual mass file must be supplied to PIFLASH as a local file assigned to logical unit 7 and referred to as TAPE7.

(4) The shell mode files must be supplied to the PIFLASH code as local files assigned to logical units selected from the set 10,11,12,13,14,15,20. These files must be referred to as TAPE10, TAPE11, etc. Additional shell mode files may be accomodated by changing the PROGRAM card of the PIFLASH code through the use of UPDATE.

(5) When the problem under consideration includes internal appendages, the user should supply control cards or control statements to ensure that the shell-fluid file (TAPE99 on logical unit 99) is comprised of system-logical records. This allows simple control card operations to be used to append data for internal appendages to the shell-fluid file. If system-logical records are not used, an additional processor must be furnished to read and transfer shell, fluid, and substructure data to a single composite file.

(6) File TAPE99 must be saved to provide data needed subsequently.

3. Description of Input Data

The set of input data required for an execution of the PIFLASH code is described below. Following Appendix B, the input data specifications are written in a style similar to FORTRAN.

Input Set 1 FORMAT (5I5)

NUMBER, NTORSN, NPTM, NSYMS, NSYMP

where NUMBER = total number of circumferential harmonics to be processed, exclusive of the circumferential harmonic associated with the torsional and rolling modes of the shell S.

NTORSN = integer flag indicating whether or not the torsional and rolling modes of the shell S are to be processed (1=yes; 0=no).

NPTM = number of concentrated or point masses attached to the shell S.

NSYMS = integer flag indicating whether or not symmetry is employed about the longitudinal midpoint of the shell S (1=yes; 0=no).

NSYMP = integer flag indicating whether or not symmetry is to be taken into account for the point masses (1=yes; 0=no). If NSYMP = 1, all weights of point masses supplied to PIFLASH will be doubled prior to the formation of the equations of motion. For the current version of PIFLASH, use NSYMP = NSYMS.

Input Set 2 FORMAT (4I5)

(NWETSG(K), NUSESG(K), K = 1, 2)

where NWETSG(K) = number of wet segments in the type K model of the shell of revolution S.

NUSESG(K) = number of segments used from the total number available in the type K model of the shell of revolution S [NUSESG(K) $\geq$ NWETSG(K)].
K = 1 : input corresponds to the full model of the shell S.
K = 2 : input corresponds to the compartment model of the shell S if the approximate modeling technique was used. If the straightforward modeling technique was used, the input corresponds to the full model of the shell S (i.e., same values supplied for K=1 and K=2).

Note: The components of the modes of the shell S are written on file TAPE99 only for the first NUSESG(K) segments for each K [i.e., for segments 1,2,..., NUSESG(K)]. If a substructure is attached to a given part of the shell S, the appropriate segments must be included for both K=1 and K=2. If a point mass is attached to a given part of the shell S, the appropriate segment must be included for K=1 and, if it exists, for K=2. A segment may not exist for K=2 because it falls outside of the region of interest when the approximate modeling technique is used. If a segment is included for K=1, the corresponding segment must be included for K=2, if it exits, to allow subsequent production of velocity-time histories.

Input Set 3 FORMAT (E10.0)

GRAVTY

where GRAVTY = acceleration due to gravity.

Input Set 4 FORMAT (15I5)

(NTAPE(J), J = 1, NITEMS)

where NITEMS = total number of circumferential harmonics to be processed, inclusive of the circumferential harmonic associated with the torsional and rolling modes if such modes are active in the problem under study.
NTORSN = 0 : NITEMS = NUMBER.
NTORSN = 1 : NITEMS = NUMBER + 1.

NTAPE = array containing the logical unit numbers on which the shell mode files are supplied to PIFLASH. If torsional and rolling modes are employed, the logical unit number furnished for these modes must be the last element of the NTAPE array.

Note: The logical unit numbers in the NTAPE array must be supplied in the order corresponding to N=0 (breathing), N=1, N=2,..., N=NUMBER-1, N=0 (torsional and rolling). If the torsional and rolling modes are not active, only the first NUMBER elements of the NTAPE array must be supplied.

Input Set 5 FORMAT (15I5)

(NJUSE(J), J = 1, NITEMS)

where NJUSE = array in which each element indicates the number of modes of the shell S to be used for a given circumferential harmonic N. In the NJUSE array, the number of modes used for each value of N must correspond in order to the logical unit number supplied in the NTAPE array for the same value of N.

Input Set 6 FORMAT (15I5)

(KORSG(K), K = 1, NKORSG)

where NKORSG = size of segment correspondence array KORSG. If the approximate modeling technique was used for the shell S, then NKORSG is the number of segments in the compartment model. If the straightforward modeling technique was used for the shell S, then NKORSG is the number of segments in the full mathematical model.

KORSG(K) = segment in full model of the shell S which corresponds to segment K in the compartment model. If straightforward modeling was used for the shell S, supply the values 1,2,...,NSEG, where NSEG is the total number of segments in the full mathematical model.

Input Set 7 FORMAT (15I5)

```
   DO 30 K = 1, NITEMS
30 READ (JUSE(J,K), J = 1, NJUSE(K))
```

where JUSE = array containing the identification numbers of the modes of the shell S selected for use by the user. The number of modes used and the circumferential harmonic N associated with a given column K of the JUSE array must correspond to element K of the NJUSE array. Within each column K of the JUSE array, the identification numbers must appear in increasing order.

Input Set 8 FORMAT (E10.0,2I5,E10.0)

(WEIGHT(J), LSEGM(J), LMESH(J), ANGPTM(J), J = 1, NPTM)

where WEIGHT(J) = weight of the J-th point or concentrated mass attached to the shell of revolution S.
LSEGM(J) = segment of the shell S in which the J-th point mass is attached.
LMESH(J) = mesh point in segment LSEGM(J) of the shell S at which the J-th point mass is attached.
ANGPTM(J) = circumferential coordinate angle θ (in degrees) at which the J-th point mass is attached.

Note 1: If no concentrated masses are attached to the shell S, skip Input Set 8.

Note 2: Physical points on the shell S are located with reference to a full mathematical model in accordance with the conventions established for the BOSOR4 code. Thus, to locate a given physical point on the shell S, the user must furnish (1) the number of a segment, (2) the number of a mesh or nodal point within the given segment, and (3) a circumferential coordinate angle (Fig. 1). The nodal points are actually points on the reference meridian used to generate the shell of revolution S. They are numbered, starting from unity, within each segment, and may be thought of as meridional stations.

Note 3: For side-on or oblique loading, the origin of the circumferential coordinate angle θ (or equivalently, the direction of the positive Z-axis) is determined by the point at which the shock wave first contacts the shell S, as indicated in Fig.1. For end-on loading, in which the shock wave first contacts the pole of an end closure, the origin of the circumferential coordinate angle θ is selected by the user. In this case, a horizontal Z-axis is the recommended configuration.

Note 4: The number of logical records written on the shell-fluid file by the PIFLASH code (NSF) is given by the expression

NSF = 29 + 5*NITEMS + 2*NUMBER + NEXTRA

in which NEXTRA = 1 if NPTM > 0

NEXTRA = 0 if NPTM $\leq$ 0.

4. Sample Control Cards

A sample set of job control language statements or control cards for an execution of PIFLASH under a NOS 1 operating system is shown below. A brief description of these control statements is presented in the next subsection. Various other arrangements of the job steps are possible.

```
PIFLASH(T177) RANLET
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(777)
ATTACH(OLDPL=PIFLASH)
UPDATE(F,L=A124)
RETURN(OLDPL)
FTN(I=COMPILE,L=0,OPT=1,A)
RETURN(COMPILE)
GET(TAPE7=VIRMAS)
ATTACH(TAPE10=BOSSY00)
ATTACH(TAPE11=BOSSY01)
ATTACH(TAPE12=BOSSY02)
ATTACH(TAPE13=BOSSY03)
FILE(TAPE99,RT=S,BT=C)
PURGE(SHLFLU/NA)
DEFINE(TAPE99=SHLFLU/CT=S,M=R)
RFL(70000)
LDSET(PRESET=INDEF,MAP=SB)
LDSET(FILES=TAPE99)
LGO.
GOTO(CATALOG)
EXIT.
CATALOG(TAPE99,R)
```

5. Description of Control Cards

```
PIFLASH(T177) RANLET
```
Job card or job statement.
```
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(777)
```
System dependent accounting information.
```
ATTACH(OLDPL=PIFLASH)
```
Assign permanent file PIFLASH to current job with local file name OLDPL. File PIFLASH contains the UPDATE program library for the PIFLASH code.
```
UPDATE(F,L=A124)
RETURN(OLDPL)
```
Process UPDATE directives supplied in the job stream (on file INPUT) and prepare a source file suitable for compilation.
```
FTN(I=COMPILE,L=0,OPT=1,A)
RETURN(COMPILE)
```
Produce compiled code or object module for the PIFLASH code.
```
GET(TAPE7=VIRMAS)
```
Retrieve copy of indirect access permanent file VIRMAS for use as local file TAPE7. File VIRMAS is the virtual mass file.
```
ATTACH(TAPE10=BOSSY00)
ATTACH(TAPE11=BOSSY01)
ATTACH(TAPE12=BOSSY02)
ATTACH(TAPE13=BOSSY03)
```
Assign permanent files BOSSY00, BOSSY01, BOSSY02, BOSSY03 to current job with local file names TAPE10, TAPE11, TAPE12, TAPE13, respectively. These files are the shell mode files.
```
FILE(TAPE99,RT=S,BT=C)
```
Specify the record type of local file TAPE99. This assures that the appropriate system routines are loaded for the processing of local file TAPE99. For the system-logical records indicated (RT=S), each record occupies an integral number of central memory words and is terminated by a system-supplied terminating marker. The block type indicated (BT=C) is not applicable to S-type records, but is provided for compatibility with SCOPE2.
```
PURGE(SHLFLU/NA)
DEFINE(TAPE99=SHLFLU/CT=S,M=R)
```
Assign space on permanent file device for the shell-fluid file (TAPE99) under the permanent file name SHLFLU. Information written on TAPE99 during the execution of PIFLASH is written directly on the permanent file.
```
RFL(70000)
```
Request field length for execution. This is optional on many systems.
```
LDSET(PRESET=INDEF,MAP=SB)
LDSET(FILES=TAPE99)
LGO.
```
Select load options, load, and execute the compiled version of PIFLASH. The required set of input data is supplied in the job stream (on file INPUT). The FILES parameter appearing on the LDSET control statement above must be supplied for each file name referenced on a FILE control statement.

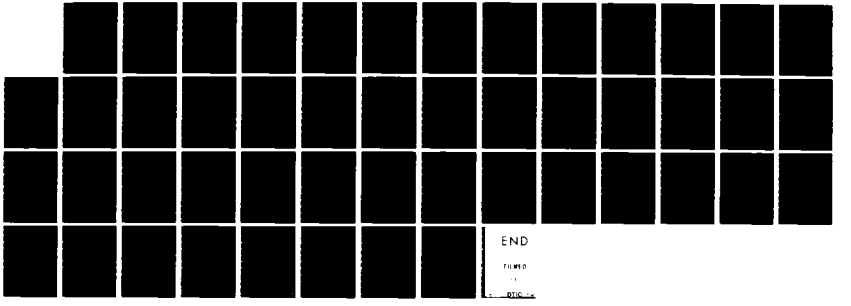
```
GOTO(CATALOG)
```
Unconditional transfer to control statement CATALOG if the execution terminates normally.

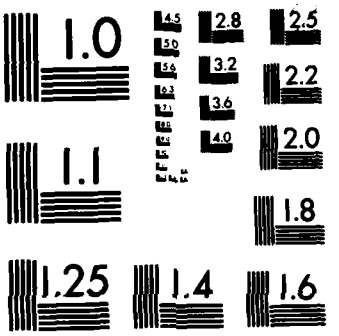
```
EXIT.
```
Terminate execution or provide transfer of control when an execution error occurs.

```
CATALOG(TAPE99,R)
```
List information about each logical record of file TAPE99 on file OUTPUT.

AD-A131 282 SUBMERGED SHOCK RESPONSE OF A LINEARLY ELASTIC SHELL OF REVOLUTION CONTAI.. (U) WEIDLINGER ASSOCIATES NEW YORK R VASUDEVAN ET AL. 01 MAY 82 DNA-TR-81-184 2/2

UNCLASSIFIED DNA001-81-C-0048 F/G 20/11 NL





6. UPDATE Directives and Input Data for Sample Problem 1

The UPDATE directives employed to prepare the shell-fluid file for Sample Problem 1 are listed below.

```
*IDENT,STORAGE
*DELETE,PIFLASH.22,PIFLASH.23
      COMMON A(10000)
      NWORDA = 10000
```

The size of the dynamic array A shown above is in excess of the 6128 words actually required for Sample Problem 1. In general, a dynamic array dimensioned to 25000 words, corresponding to a central memory requirement of about 125000 words (octal), is sufficient for most problems. A field length of almost 70000 words (octal) is required for an execution of PIFLASH when the dynamic array is dimensioned as above.

The set of input data shown below directs the PIFLASH code to prepare the shell-fluid file for Sample Problem 1.

```
 4    0    4    1    1          NUMBER,NTORSN,NPTM,NSYMS,NSYMP
 2    2    2    2               (NWETSO(K),NUSESO(K),K=1,2)
  386.4                        GRAVTY
10   11   12   13               (NTAPE(J),J=1,NITEMS)
25   30   30   30               (NJUSE(J),J=1,NITEMS)
 1    2                        (KORSO(K),K=1,NKORSO)
 2    4    6    7    8    9   11   13   14   15   16   17   18   19   20 JUSE
21   22   23   24   25   26   27   28   29   30                        JUSE
 1    2    3    4    5    6    7    8    9   10   11   12   13   14   15 JUSE
16   17   18   19   20   21   22   23   24   25   26   27   28   29   30 JUSE
 1    2    3    4    5    6    7    8    9   10   11   12   13   14   15 JUSE
16   17   18   19   20   21   22   23   24   25   26   27   28   29   30 JUSE
 1    2    3    4    5    6    7    8    9   10   11   12   13   14   15 JUSE
16   17   18   19   20   21   22   23   24   25   26   27   28   29   30 JUSE
   32.0    2   12      0.0    WEIGHT(1),LSEGM(1),LMESH(1),ANGPTM(1)
   32.0    2   12    180.0    WEIGHT(2),LSEGM(2),LMESH(2),ANGPTM(2)
   30.5    2   53     90.0    WEIGHT(3),LSEGM(3),LMESH(3),ANGPTM(3)
   30.5    2   53    270.0    WEIGHT(4),LSEGM(4),LMESH(4),ANGPTM(4)
```

7. UPDATE Directives and Input Data for Sample Problem 2

The UPDATE directives employed to prepare the shell-fluid file for Sample Problem 2 are listed below.

```
*IDENT,STORAGE
*DELETE,PIFLASH.22,PIFLASH.23
      COMMON A(10000)
      NWORDA = 10000
```

The size of the dynamic array A shown above is somewhat in excess of the size actually required for Sample Problem 2. When the dynamic array is dimensioned to 10000 words, as above, a field length of almost 70000 words (octal) is required for an execution of the PIFLASH code.

The set of input data shown below directs the PIFLASH code to prepare the shell-fluid file for Sample Problem 2.

```
 4     0     4     1     1          NUMBER,NTORSN,NPTM,NSYMS,NSYMP
 3     3     1     1                (NWETS0(K),NUSES0(K),K=1,2)
  386.4                            GRAVTY
10    11    12    13                (NTAPE(J),J=1,NITEMS)
15    30    15    15                (NJUSE(J),J=1,NITEMS)
 3                                  (KORS0(K),K=1,NKORS0)
 2     3     5     6     7     8     9    10    11    12    14    15    16    17    18 JUSE
 1     2     3     4     5     6     7     8     9    10    11    12    13    14    15 JUSE
16    17    18    19    20    21    22    23    24    25    26    27    28    29    30 JUSE
 1     2     3     4     5     6     7     8     9    10    11    12    13    14    15 JUSE
 1     2     3     4     5     6     7     8     9    10    11    12    13    14    15 JUSE
   32.0     2    11       0.0     WEIGHT(1),LSEGM(1),LMESH(1),ANGPTM(1)
   32.0     2    11     180.0     WEIGHT(2),LSEGM(2),LMESH(2),ANGPTM(2)
   30.5     3    19      90.0     WEIGHT(3),LSEGM(3),LMESH(3),ANGPTM(3)
   30.5     3    19     270.0     WEIGHT(4),LSEGM(4),LMESH(4),ANGPTM(4)
```

APPENDIX D - THE USE OF SAPIV

1. Background

The SAPIV finite-element code of Ref. [8] is included in the ELSHOK suite of computer programs for the modal analysis of the internal appendages or substructures σ. The general-purpose SAPIV code is based upon a lumped mass formulation and applies to linearly elastic structures. Comprehensive documentation of SAPIV may be found in Ref. [8]. A knowledge of the information given in Ref. [3] and Sections II and III of the present report will aid in the understanding of this appendix.

A number of modifications have been made to SAPIV to enable it to function within the scope of an ELSHOK calculation. Thus, in addition to determining the fixed-base modes and corresponding natural frequencies of a substructure, the unconstrained or free-free physical mass and stiffness matrices ($\underline{M}$ and $\underline{K}$, respectively) are also determined. These matrices are required for the calculation of the constraint modes $\hat{g}_{ji}$ of Eq.(10) and of the coefficients needed to evaluate the interaction forces and moments developed between each substructure σ and the main body S. It should be pointed out that the entire physical mass and stiffness matrices of a substructure are not required for the subsequent submerged shock response problem. In fact, as discussed in Ref. [3], the solution of this response problem requires only those rows of $\underline{K}$ and and only those main diagonal terms of $\underline{M}$ which correspond to the physical degrees of freedom of σ constrained to move with S.

2. Rules of Operation

For compatibility with the current version of the ELSHOK code, the following rules must be observed when executing the SAPIV code:

(1) The rectangular Cartesian coordinate system (x,y,z) used to describe a substructure σ must be a right-handed system, and the x-axis of this system of coordinates must coincide in direction and sense with the longitudinal or X-axis of the main body S.

(2) If a substructure is symmetrical about the longitudinal midpoint of S, in a problem for which symmetry is being used, the user must model only the left half of the substructure. If symmetry is not applicable, the entire model of such a substructure must be provided.

(3) The input data supplied for SAPIV in an ELSHOK application must be provided in two parts. The first part is standard input prepared in accordance with the requirements of Ref. [8], except that (1) additional input items must be supplied on the master control and dynamic analysis input cards and (2) the physical degrees of freedom associated with the supports or bases of the substructure under study must not be constrained. The second part, appended to the first part without a separator of any kind, supplies the fixity allowing the calculation of the fixed-base modes and corresponding natural frequencies.

(4) A separate execution of SAPIV must be made for each substructure included in an analysis.

3. Modifications to SAPIV

A number of changes have been made to SAPIV to facilitate its use in the ELSHOK suite of computer programs. A brief description of each of these modifications follows:

(1) A file has been supplied and assigned to logical unit 98. This file is referred to by the local file name TAPE98 and serves as a scratch file.

(2) A file has been supplied and assigned to logical unit 99. This file is referred to by the local file name TAPE99 and serves as the substructure mode file of Section III. File TAPE99 contains geometry, natural frequencies, fixed-base modes, and the unconstrained physical mass and stiffness matrices, in unformatted binary records, for a given substructure. The substructure mode file must be saved after each execution of SAPIV to provide data needed subsequently.

(3) Although not a modification to SAPIV, it should be pointed out here that a small plotting processor (PLOTSAM) is supplied with ELSHOK. This processor prepares plots on a TEKTRONIX graphics terminal and may be used to plot the finite-element model and the fixed-base modes of a given substructure.

(4) Additional input items have been added to the usual input data. These items were mentioned in the preceding subsection and will be described in detail in the next subsection.

The above modifications have been made through the use of the UPDATE batch editing system (Ref. [9]) and may be easily removed to return SAPIV to its standard configuration.

4. Additional Input

When using SAPIV with the ELSHOK code, all of the applicable input data described in Ref. [8] must be prepared for the problem of interest in accordance with the above rules of operation. Thus, two additional input parameters (NRIGID and SHIFT) must be supplied within the standard input data. Descriptions of these parameters are given below. Following Appendix B, the input data specifications are written in a style similar to FORTRAN.

For ELSHOK applications, the master control card in the standard SAPIV input data (Input Section II of Ref. [8]) has been changed to

Master Control Card — FORMAT (10I5)

```
NUMNP, NELTYP,  LL,   NF,    NDYN,
MODEX, NAD,     KEQB, N10SV, NRIGID
```

where NUMNP = total number of nodal points in the model.
NELTYP = number of element groups or types.
LL = number of loading conditions. For ELSHOK applications, use LL = 0.
NF = number of frequencies to be found.
NDYN = analysis type code. For ELSHOK applications, use NDYN = 1.
MODEX = program execution mode. For ELSHOK applications, use use MODEX = 0.
NAD = total number of vectors to be used in a subspace iteration solution for eigenvalues and eigenvectors. For ELSHOK applications, set NAD = 0 and accept the default value.
KEQB = number of degrees of freedom (equations) per block of storage. For ELSHOK applications, set KEQB = 0 and accept the value calculated automatically by SAPIV.
N10SV = integer flag indicating whether or not stress information is to be written on local file TAPE10 for post-processing. For ELSHOK applications, use N10SV = 0.
NRIGID = integer flag indicating whether or not free-free modes, including rigid body modes, are to be determined (1=yes; 0=no). For the current version of ELSHOK, which employs fixed-base modes, use NRIGID = 0.

The dynamic analysis card used to set parameters for the calculation of mode shapes and frequencies in the standard SAPIV input data (Input Section VII A of Ref. [8]) has been changed to

Dynamic Analysis Card FORMAT (3I5,2E10.0,I5,E10.0)

IFPR, IFSS, NITEM, RTOL, COFQ, NFO, SHIFT

where IFPR = flag for printing intermediate output (1=yes; 0=no). For ELSHOK applications, use IFPR = 0.
IFSS = flag for by-passing the Sturm sequence check in an eigenvalue calculation (1=yes; 0=no).
NITEM = maximum number of iterations allowed for reaching the convergence tolerance. For ELSHOK applications, set NITEM = 0 and accept the default value.
RTOL = convergence tolerance for the highest frequency computed. For ELSHOK applications, set RTOL = 0.0 and accept the default value.
COFQ = cutoff frequency in cycles per unit time. For ELSHOK applications, set COFQ = 0.0 to allow the computation of NF frequencies.
NFO = number of starting iteration vectors. For ELSHOK applications, use NFO = 0.
SHIFT = frequency shift used in the calculation of free-free modes (NRIGID = 1). For the current version of ELSHOK, use SHIFT = 0.0.

The set of input data supplied to provide the fixity needed for the calculation of the fixed-base modes and corresponding natural frequencies must be placed immediately after the dynamic analysis card described above. A description of this additional input follows:

Input Set 1 FORMAT (2I5)

NFACE, NSYMTY

where NFACE = number of interface or attachment points between the substructure σ under study and the shell S.

NSYMTY = integer flag indicating whether or not symmetry is employed in the finite element model of the substructure (1=yes; 0=no). For the current version of ELSHOK, only symmetry associated with the x-direction (i.e., about a y-z plane) may be incorporated into the model.

Note 1 : For the current version of ELSHOK, symmetry may not be employed unless the entire structure under consideration (including the shell S, any internal appendages σ, and any concentrated masses) is symmetrical about the longitudinal midpoint on the main body S.

Note 2 : When NSYMTY = 1, all effects of a substructure on the shell S are doubled for use in the equations governing the submerged shock response problem. Thus, in a problem for which symmetry is employed, identical substructures, one in the left half of S and one in the right, may be accounted for by modeling the left-hand structure without the use of symmetry and by setting NSYMTY = 1.

Input Set 2 FORMAT (7I5)

```
   DO 40 K = 1, NFACE
40 READ NODE, (NFIXTY(J), J = 1, 6)
```

where NODE = identification number of a nodal point serving as an interface or attachment point between the substructure σ under study and the shell S.

NFIXTY = array containing the boundary condition codes (1 or 0) for nodal point NODE. Elements 1-6 of this array correspond to the x-translation, y-translation, z-translation, x-rotation, y-rotation, and z-rotation boundary condition codes, respectively, for the indicated nodal point.

NFIXTY(J) = 1: physical degree of freedom J is constrained to move with S or is not active in the problem.

NFIXTY(J) = 0: physical degree of freedom J is not constrained to move with S but is active.

Note : In the terminology of Ref.[8], slave degrees of freedom may not be constrained to move with the main body S.

The two sets of input data described above must be followed by the usual blank or dummy title and master control cards used to terminate an execution of SAPIV.

5. Sample Control Cards

Shown below is a typical set of control cards or job control language statements for an execution of SAPIV under a NOS 1 operating system. A brief description of these control statements is given in the next subsection. Various other arrangements of the job steps are possible.

```
SAPIV(T177) RANLET
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(777)
ATTACH(LGO=SAPFTN)
FILE(TAPE1,BT=C,RT=S)
FILE(TAPE2,BT=C,RT=S)
FILE(TAPE3,BT=C,RT=S)
FILE(TAPE4,BT=C,RT=S)
FILE(TAPE7,BT=C,RT=S)
FILE(TAPE8,BT=C,RT=S)
FILE(TAPE9,BT=C,RT=S)
FILE(TAPE10,BT=C,RT=S)
FILE(TAPE98,BT=C,RT=S)
FILE(TAPE99,BT=C,RT=S)
PURGE(SUBMODE/NA)
DEFINE(TAPE99=SUBMODE/CT=S,M=R)
RFL(170000)
LDSET(PRESET=INDEF,MAP=N)
LDSET(FILES=TAPE1/TAPE2/TAPE3/TAPE4/TAPE7)
LDSET(FILES=TAPE8/TAPE9/TAPE10/TAPE98/TAPE99)
LGO.
GOTO(CATALOG)
EXIT.
CATALOG(TAPE99,R)
```

6. Description of Control Cards

```
SAPIV(T177) RANLET
```
Job card or job statement.
```
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(777)
```
System dependent accounting information.
```
ATTACH(LGO=SAPFTN)
```
Assign permanent file SAPFTN to current job with local file name LGO. File SAPFTN contains the compiled code or object module for SAPIV.
```
FILE(TAPE1,BT=C,RT=S)
FILE(TAPE2,BT=C,RT=S)
FILE(TAPE3,BT=C,RT=S)
FILE(TAPE4,BT=C,RT=S)
FILE(TAPE7,BT=C,RT=S)
FILE(TAPE8,BT=C,RT=S)
FILE(TAPE9,BT=C,RT=S)
FILE(TAPE10,BT=C,RT=S)
FILE(TAPE98,BT=C,RT=S)
FILE(TAPE99,BT=C,RT=S)
```
Specify the record type of each of the indicated local files. This assures that appropriate system routines are loaded for processing.
```
PURGE(SUBMODE/NA)
DEFINE(TAPE99=SUBMODE/CT=S,M=R)
```
Assign space on permanent file device for the substructure mode file (TAPE99) under the permanent file name SUBMODE. Information written on TAPE99 during the execution of SAPIV is written directly on the permanent file.
```
RFL(170000)
```
Request field length for execution. This is optional on many systems.
```
LDSET(PRESET=INDEF,MAP=N)
LDSET(FILES=TAPE1/TAPE2/TAPE3/TAPE4/TAPE7)
LDSET(FILES=TAPE8/TAPE9/TAPE10/TAPE98/TAPE99)
LGO.
```
Select load options, load, and execute the compiled version of SAPIV. The required set of input data is supplied in the job stream (on file INPUT).
```
GOTO(CATALOG)
```
Unconditional transfer to control statement CATALOG if the execution terminates normally.
```
EXIT.
```
Terminate execution or provide transfer of control when an execution error occurs.
```
CATALOG(TAPE99,R)
```
List information about each logical record of file TAPE99 on file OUTPUT. The use of this control statement is not recommended for finite element calculations involving more than about 100 physical degrees of freedom.

7. Input Data for Sample Problem 1

The set of input data shown below directs the SAPIV code, as modified for use with ELSHOK, to perform the modal analysis of the substructure included in Sample Problem 1.

```
SYMMETRICAL SUBSTRUCTURE -- 05/15/82
  25    1    0    5    1    0    0    0    0    0 NUMNP,NELTYP,....,N10SV,NR16ID
   1    1    1    0    1    1    1        20.0        0.0 -16.71875    0
   5    1    1    0    1    1    1        20.0        0.0 -3.343750    1
   6    1    1    0    1    0    1         0.0        0.0         0.0    0
  15    1    1    0    1    0    1        20.0        0.0         0.0    1
  16    1    1    0    1    0    1    22.48125        0.0         0.0    0
  24    1    1    0    1    0    1    42.33125        0.0         0.0    1
  25    1    1    0    1    1    1     44.8125        0.0         0.0    0
   2   24    2    0    2
   1 29.0E 06        0.3  7.29E-04
   2 29.0E 06        0.3       0.0
   1       8.0       0.0       0.0        1.0 2.6666667 10.666667
   2       6.0       0.0       0.0        1.0      18.0        0.5
       0.0         0.0         0.0         0.0                                MULT G X
       0.0         0.0         0.0         0.0                                MULT G Y
       0.0         0.0         0.0         0.0                                MULT G Z
   1    1    2   25    2    2
   4    4    5   25    2    2
   5    5   15   25    2    2                          000001
   6    6    7    1    1    1
  24   24   25    1    1    1
   0    0         0.0         0.0         0.0         0.0         0.0         0.0 CONC MASS
      0.0         0.0         0.0         0.0    ELEMENT LOAD MULTIPLIERS
   0    0    0         0.0         0.0    0         0.0         IFPR,IFSS,....,NFO,SHIFT
   1    1                                NFACE,NSYMTY
   1    1    1    1    1    1    1                    NODE,(NFIXTY(J),J=1,6)
END OF DATA (DUMMY TITLE)
   0    0    0    0    0    0    0    0    0    0
```

8. Input Data for Sample Problem 2

The set of input data shown below directs the SAPIV code, as modified for use with ELSHOK, to perform the modal analysis of the substructure included in Sample Problem 2.

```
SYMMETRICAL SUBSTRUCTURE -- 05/15/82
   25     1     0     5     1     0     0     0     0     0 NUMNP,NELTYP,....,N10SV,NRIGID
    1     1     1     0     1     1     1          20.0          0.0 -16.71875     0
    5     1     1     0     1     1     1          20.0          0.0 -3.343750     1
    6     1     1     0     1     0     1           0.0          0.0          0.0     0
   15     1     1     0     1     0     1          20.0          0.0          0.0     1
   16     1     1     0     1     0     1   22.48125          0.0          0.0     0
   24     1     1     0     1     0     1   42.33125          0.0          0.0     1
   25     1     1     0     1     1     1    44.8125          0.0          0.0     0
    2    24     2     0     2
    1  29.0E 06          0.3   7.29E-04
    2  29.0E 06          0.3         0.0
    1         8.0         0.0         0.0          1.0 2.6666667 10.666667
    2         6.0         0.0         0.0          1.0      18.0         0.5
        0.0           0.0           0.0           0.0                                MULT G X
        0.0           0.0           0.0           0.0                                MULT G Y
        0.0           0.0           0.0           0.0                                MULT G Z
    1     1     2    25     2     2
    4     4     5    25     2     2
    5     5    15    25     2     2                          000001
    6     6     7     1     1     1
   24    24    25     1     1     1
    0     0           0.0           0.0           0.0          0.0          0.0          0.0 CONC MASS
        0.0           0.0           0.0           0.0    ELEMENT LOAD MULTIPLIERS
    0     0     0           0.0           0.0     0          0.0          IFPR,IFSS,....,NFO,SHIFT
    1     1                                     NFACE,NSYMTY
    1     1     1     1     1     1     1                NODE,(NFIXTY(J),J=1,6)
END OF DATA (DUMMY TITLE)
    0     0     0     0     0     0     0     0     0     0
```

The above set of input data is identical to that employed for Sample Problem 1. Thus, the execution of SAPIV for Sample Problem 2 may be skipped if the substructure mode file for Sample Problem 1 is available for the processing required in the solution of Sample Problem 2.

APPENDIX E - THE USE OF PICRUST

1. General Information

The PICRUST computer code is a post-processor for the SAPIV code. As discussed in Section III, the PICRUST code takes data from a substructure mode file (produced by SAPIV), adds to it, and reorganizes it to facilitate the solution of the equations governing the transient response problem under study. During an execution of PICRUST, connectivity between a substructure σ and the shell S is accounted for, constraint modes are computed, and coefficients for evaluating the forces developed at the points of attachment between σ and S are determined. Familiarity with Appendix D and Ref. [3] will aid in the understanding of this appendix.

2. Rules of Operation

(1) A substructure mode file from an execution of SAPIV must be available prior to the execution of the PICRUST code.

(2) A separate execution of PICRUST is required for each substructure included in the problem under consideration.

(3) The substructure mode file must be supplied to PICRUST as a local file assigned to logical unit 4 and referred to as TAPE4.

(4) For ease of merging files to complete the input file required for the submerged shock response problem, the user should supply control cards or control statements to ensure that the substructure file produced by PICRUST (TAPE10 on logical unit 10) is comprised of system-logical records.

(5) File TAPE10 must be saved to provide data needed subsequently.

3. Description of Input Data

The input data specifications required for an execution of the PICRUST code are described below. Following Appendix B, these specifications are written in a style similar to FORTRAN.

Input Set 1 FORMAT (10I5)

(LISTIT(J), J = 1, 20)

where LISTIT(1) = output flag for boundary condition codes of free-free or unconstrained substructure.
LISTIT(2) = output flag for equation numbers or problem unknowns associated with free-free substructure.
LISTIT(3) = output flag for nodal point geometry.
LISTIT(4) = output flag for main diagonal terms of diagonal mass matrix of free-free substructure.
LISTIT(5) = output flag for stiffness matrix of free-free substructure.
LISTIT(6) = output flag for connectivity data for interface or attachment points between current substructure and main body S.
LISTIT(7) = output flag for equation numbers of free-free substructure after partitioning of physical degrees of freedom to correspond to Eqs. (9) and (10).
LISTIT(8) = output flag for main diagonal terms of diagonal mass matrix of free-free substructure after partitioning.
LISTIT(9) = output flag for stiffness matrix of free-free substructure after partitioning.
LISTIT(10) = output flag for constraint modes in blocking (by rows) used for their solution.
LISTIT(11) = output flag for constraint modes in blocking (by columns) used to display all physical degrees of freedom in a given constraint mode.
LISTIT(12) = output flag for fixed-base natural frequencies.
LISTIT(13) = output flag for fixed-base modes displayed with status of six degrees of freedom at each node shown.
LISTIT(14) = output flag for fixed-base modes displayed in compacted form (i.e., with only active degrees of freedom shown).
LISTIT(15) = output flag for expansion coefficients of a modal series representation, in terms of fixed-base modes of substructure, of constraint modes.
LISTIT(16) = output flag for skyline bandwidth of those rows of partitioned free-free stiffness matrix corresponding to physical degrees of freedom of substructure constrained to move with shell S.

LISTIT(17) = output flag for portion of above rows of partitioned free-free stiffness matrix [LISTIT(16)] corresponding to physical degrees of freedom of substructure not constrained to move with shell S.

LISTIT(18) = output flag for interface force coefficients associated with fixed-base modes of substructure.

LISTIT(19) = output flag for portion of above rows of partitioned free-free stiffness matrix [LISTIT(16)] corresponding to physical degrees of freedom of substructure constrained to move with shell S.

LISTIT(20) = output flag for interface force coefficients associated with constraint modes of substructure.

Note 1: If an output flag is set to 1, output is supplied. If an output flag is set to 0, no output is supplied.

Note 2: In almost all cases, use LISTIT(5) = LISTIT(9) = LISTIT(17) = LISTIT(19) = 0, since the structural stiffness matrix is a very large array for most problems of interest.

Input Set 2 FORMAT (3I5)

NJUSE, NHWSOB, NHWBAR

where NJUSE = number of fixed-base modes of substructure to be used for subsequent response calculation. The value of NJUSE must not exceed the value of NF used in the SAPIV calculation.

NHWSOB = number of hectowords defining size of storage blocks used in calculation of constraint modes of substructure.
NHWSOB $\leq$ 0: constraint modes computed in one block (entire array in core), if possible.
NHWSOB >0: constraint modes computed in blocks of specified or smaller size, depending on available storage.

NHWBAR = number of hectowords defining size of storage blocks used for interface force coefficients associated with constraint modes of substructure. For the current version of ELSHOK, use NHWBAR = 0. This specification directs the PICRUST code to write the entire array of coefficients on the substructure file in one block.

Note: If the size of the constraint mode array is smaller than about 10000 words, the use of NHWSOB = 0 is recommended. The constraint mode array has NUDOF rows and NCDOF columns, where NUDOF is the number of physical degrees of freedom of σ not constrained to move with S and NCDOF is the number of physical degrees of freedom of σ constrained to move with S.

INPUT Set 3 FORMAT (3I5,5X,E10.0)

```
   DO 50 K = 1, NFACE
50 READ NIPSUB, LBOSEG, LBOSPT, ANGDEG
```

where NIPSUB = node number of substructure σ which is connected to the shell S at the K-th interface or attachment point between σ and S.

LBOSEG = segment of the shell S in which the K-th interface point is located.

LBOSPT = mesh point in segment LBOSEG of the shell S at which the K-th connection between σ and S is made.

ANGDEG = circumferential coordinate angle θ (in degrees) at which the K-th interface point is located (Fig.1).

Note 1 : The number of interface points (NFACE) is retrieved from the substructure mode file and need not be supplied by the user during an execution of PICRUST.

Note 2: The individual sets of the above four connection parameters may be supplied to the PICRUST code in any convenient order.

Note 3: Physical points on the shell S are located with reference to a full mathematical model in accordance with the conventions established for the BOSOR4 code.

Input Set 4 FORMAT (E10.0)

```
DEGROT
```

where DEGROT = orientation angle (in degrees) of coordinate axes of substructure σ with respect to global coordinate axes of shell S. The angle DEGROT (α_σ of Fig. 1) defines the angle between the positive z-axis of σ and the positive Z-axis of S. Positive angles are measured counterclockwise (when looking in the positive X-direction) from the Z-axis of S to the z-axis of σ.

Input Set 5 FORMAT (15I5)

(JUSE(J), J = 1, NJUSE)

where JUSE = array containing the identification numbers of the fixed-base modes of substructure σ selected for use by the user.

Note: The number of logical records written on the substructure file by the PICRUST code (NSB) is given by the expression

NSB = 15 + NBLOCK + NBLFOR

in which NBLOCK is the number of storage blocks needed for the constraint modes and NBLFOR = 1 for the current version of ELSHOK.

4. Sample Control Cards

A sample set of job control cards or job control language statements for an execution of PICRUST under a NOS 1 operating system is shown below. The next subsection contains a brief description of these control statements. Various other arrangements of the job steps are possible.

```
PICRUST(T277) RANLET
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(777)
ATTACH(OLDPL=PICRUST)
UPDATE(F,L=A124)
RETURN(OLDPL)
FTN(I=COMPILE,L=0,OPT=1,PL=25000,A)
RETURN(COMPILE)
FILE(TAPE4,BT=C,RT=S)
FILE(TAPE6,BT=C,RT=S)
FILE(TAPE7,BT=C,RT=S)
FILE(TAPE8,BT=C,RT=S)
FILE(TAPE9,BT=C,RT=S)
FILE(TAPE10,BT=C,RT=S)
ATTACH(TAPE4=SUBMODE)
PURGE(SYMSUB/NA)
DEFINE(TAPE10=SYMSUB/CT=S,M=R)
RFL(110000)
LDSET(PRESET=INDEF,MAP=SB)
LDSET(FILES=TAPE4/TAPE10)
LDSET(FILES=TAPE6/TAPE7/TAPE8/TAPE9)
LGO.
GOTO(CATALOG)
EXIT.
CATALOG(TAPE10,R)
```

5. Description of Control Cards

```
PICRUST(T277) RANLET
```
Job card or job statement.
```
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(777)
```
System dependent accounting information.
```
ATTACH(OLDPL=PICRUST)
```
Assign permanent file PICRUST to current job with local file name OLDPL. File PICRUST contains the UPDATE program library for the PICRUST code.
```
UPDATE(F,L=A124)
RETURN(OLDPL)
```
Process UPDATE directives supplied in the job stream (on file INPUT) and prepare a source file suitable for compilation.
```
FTN(I=COMPILE,L=0,OPT=1,PL=25000,A)
RETURN(COMPILE)
```
Produce compile code or object module for the PICRUST code.
```
FILE(TAPE4,BT=C,RT=S)
FILE(TAPE6,BT=C,RT=S)
FILE(TAPE7,BT=C,RT=S)
FILE(TAPE8,BT=C,RT=S)
FILE(TAPE9,BT=C,RT=S)
FILE(TAPE10,BT=C,RT=S)
```
Specify the record type of each of the indicated local files. This assures that appropriate system routines are loaded for processing.
```
ATTACH(TAPE4=SUBMODE)
```
Assign permanent file SUBMODE to current job with local file name TAPE4. File SUBMODE is the substructure mode file.
```
PURGE(SYMSUB/NA)
DEFINE(TAPE10=SYMSUB/CT=S,M=R)
```
Assign space on permanent file device for the substructure file (TAPE10) under the permanent file name SYMSUB. Information written on TAPE10 during the execution of PICRUST is written directly on the permanent file.
```
RFL(110000)
```
Request field length for execution. This is optional on many systems.
```
LDSET(PRESET=INDEF,MAP=SB)
LDSET(FILES=TAPE4/TAPE10)
LDSET(FILES=TAPE6/TAPE7/TAPE8/TAPE9)
LGO.
```
Select load options, load,and execute the compiled version of PICRUST. The required set of input data is supplied in the job stream (on file INPUT). The FILES parameter must be supplied on a LDSET control statement for each local file name referenced on a FILE control statement.
```
GOTO(CATALOG)
```
Unconditional transfer to control statement CATALOG if the execution terminates normally.
```
EXIT.
```
Terminate execution or provide transfer of control when an execution error occurs.
```
CATALOG(TAPE10,R)
```
List information about each logical record of file TAPE10 on file OUTPUT.

6. UPDATE Directives and Input Data for Sample Problem 1

The UPDATE directives employed to prepare the substructure file for Sample Problem 1 are listed below.

```
*IDENT,STORAGE
*DELETE,PICRUST.14,PICRUST.15
      COMMON A(10000)
      NWORDA = 10000
```

The size of the dynamic array A shown above is in excess of the 2221 words actually required for Sample Problem 1. In general, a dynamic array dimensioned to 50000 words, corresponding to a central memory requirement of about 210000 words (octal), is sufficient for most problems. When the dynamic array is dimensioned to 10000 words, as above, a field length of almost 110000 words (octal) is required for an execution of the PICRUST code.

The set of input data shown below directs the PICRUST code to prepare the substructure file for Sample Problem 1.

```
1    1    1    1    0    1    1    1    0    0    OUTPUT FLAGS  1-10
0    1    1    1    1    1    0    1    0    1    OUTPUT FLAGS 11-20
5    0    0                   NJUSE,NHWSOB,NHWBAR
1    2   42          180.0    NIPSUB,LBOSEG,LBOSPT,ANGDEG
   0.0                        DEGROT
1    2    3    4    5         (JUSE(J),J=1,NJUSE)
```

7. UPDATE Directives and Input Data for Sample Problem 2

The UPDATE directives employed to prepare the substructure file for Sample Problem 2 are listed below.

```
*IDENT,STORAGE
*DELETE,PICRUST.14,PICRUST.15
      COMMON A(10000)
      NWORDA = 10000
```

The size of the dynamic array A shown above is somewhat in excess of the size actually required for Sample Problem 2. A field length of almost 110000 words (octal) is required for an execution of PICRUST when the dynamic array is dimensioned as above.

The set of input data shown below directs the PICRUST code to prepare the substructure file for Sample Problem 2.

```
1    1    1    1    0    1    1    1    0    0    OUTPUT FLAGS  1-10
0    1    1    1    1    1    0    1    0    1    OUTPUT FLAGS 11-20
5    0    0                   NJUSE,NHWSOB,NHWBAR
1    3    1         180.0     NIPSUB,LBOSEG,LBOSPT,ANGDEG
   0.0                        DEGROT
1    2    3    4    5         (JUSE(J),J=1,NJUSE)
```

APPENDIX F - THE USE OF USLOB

1. General Information

The USLOB computer code is included in the ELSHOK suite of computer programs to solve the governing equations of the structure-fluid interaction problem outlined in Section II. The USLOB time-integration processor employs the Runge-Kutta technique, in the modification due to Gill (Ref.[11]), to perform the integration in time. As discussed in Section III, the major portion of the input data required for the structure-fluid interaction calculation must be supplied in a file termed the "input file". In addition to this file, the user must also supply time step information and a description of the shock wave loading. Printed velocity-time histories may also be obtained.

2. Completion of Input File

Before the USLOB code may be applied to a submerged shock response problem, the input file of Section III must be completed. For problems without internal appendages, the shell-fluid file (Appendix C) serves as the input file. For problems involving substructures, all substructure files (Appendix E) must be appended to the shell-fluid file.*) If these files are comprised of system-logical records, the merging of the various files may be easily accomplished by means of simple control card or job control language operations, as shown below. If the shell-fluid and substructure files are not comprised of system-logical records, a FORTRAN computer program may be written to complete the input file. The use of system-logical records is the recommended procedure.

---

*) For the current version of USLOB, the substructure files may be appended to the shell-fluid file in any convenient order.

The following set of control cards or control statements may be used to complete the input file for Sample Problem 1:

```
MERGE(T77) RANLET
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
ATTACH(PART1=SHLFLU)
ATTACH(PART2=SYMSUB)
PURGE(PROBONE/NA)
DEFINE(NEWFILE=PROBONE/CT=S,M=R)
COPYBR(PART1,NEWFILE,58)
COPYBR(PART2,NEWFILE,17)
WRITEF(NEWFILE,1)
CATALOG(NEWFILE,R)
EXIT.
```

In the above job stream, the entire shell-fluid file (SHLFLU) and the entire substructure file (SYMSUB) are copied to the input file (PROBONE) on a record-by-record basis. This is accomplished by using the system routine COPYBR to copy the individual binary records to the permanent file PROBONE. The number of records to be copied may be obtained by means of the control statement CATALOG or may be computed using the expressions given in Appendices C and E. An end-of-file mark (EOF) is provided on the input file for the structure-fluid interaction problem by means of the system routine WRITEF. A similar set of control cards may be used to complete the input file for Sample Problem 2.

3. Rules of Operation

(1) The input file must be supplied to the USLOB code as a local file assigned to logical unit 4 and referred to as TAPE4.

(2) The generalized velocity file (TAPE10 on logical unit 10) must be saved if plotted velocity-time histories are to be made subsequently.

(3) If the problem under study involves a substructure or more than one substructure, the shell-substructure transformation file (TAPE5 on logical unit 5) must be saved if plots of substructure responses are to be made subsequently.

4. Description of Input Data

The input data specifications required for an execution of the USLOB code are described below. Following Appendix B, these specifications are written in a style similar to FORTRAN.

Input Set 1 FORMAT (7I5)

NTIME, NSKIP, NCHRG, NQUAD, NFINE, KOUPLE, NSUBS

where NTIME = number of time points used in the numerical integration of the governing equations. The corresponding number of time steps is NTIME-1.

NSKIP = integer parameter selecting the time interval at which the generalized velocities are saved for subsequent processing. The time-history information is saved at time points starting with the initial time point (t = 0) and incremented by NSKIP thereafter. For example, if NSKIP = 2, time-history information is saved at time points 1,3,5,etc.

NCHRG = integer parameter for selecting the representation of the incident spherical shock wave.
NCHRG = 1: empirical decaying exponential pressure-time history of Ref [6].
NCHRG = 2: discretized pressure-time history containing up to 25 pressure-time points.

NQUAD = number of rays dividing a quadrant of any transverse cross section of the shell S into a coarse (fixed) grid used to locate the shock front and to evaluate the contributions of the pressure- and velocity-time histories away from the front. For most ELSHOK applications, use NQUAD = 9.

NFINE = number of rays dividing a sector defined by NQUAD into a fine grid used to evaluate the contributions of the pressure- and velocity-time histories in the vicinity of the shock front. The grid defined by NFINE is established at the shock front and in essence moves with the front. For most ELSHOK applications, use NFINE = 11.

KOUPLE = integer parameter for selecting the manner of introducing the inertia coupling, produced by concentrated masses, into the modal equations of the shell S. If no concentrated masses are attached to S (NPTM = 0 in PIFLASH), supply any valid integer to satisfy the input format.
KOUPLE = 1 : full (exact) coupling of the modes of S.
KOUPLE = 2 : partial (approximate) coupling of the modes of S.

NSUBS = number of substructures included in the analysis.

Note 1 : To avoid loss of the generalized velocities at the final time point, the parameter NTIME must be selected in conjunction with the parameter NSKIP such that the calculation ends on a time point for which the generalized velocities are saved.

Note 2 : When KOUPLE = 2, simplifications based upon engineering judgment are employed in the representation of the inertia coupling, due to concentrated masses, of the modes of the shell S, leading to reduced central memory requirement and computational time for a given problem. Specifically, when KOUPLE = 2, full inertia coupling between the N=0 torsional and rolling modes and the N=1 rigid body and whipping modes of the shell S is included in the analysis. However, the inertia coupling between the N=0 breathing modes and the modes having N≥2 with the above sets of modes is neglected. Inertia coupling is included between the modes in each of the groups of modes N=0 (breathing), 2, 3, ..., but no inertia coupling between these individual groups is considered. This partial coupling has been found to yield acceptable results when used in conjunction with the simplified modeling technique of Appendix A (full and compartment models of S) for problems in which localized side-on loading acts upon the portion of S corresponding to the compartment model.

Note 3 : Since USLOB utilizes main memory management, it is quite easy to reduce the core requirement or central memory to the minimum size required for a given execution. This may be accomplished by specifying a zero or negative value of NTIME in an otherwise complete set of input data. An execution of USLOB with this set of input data will then supply information from which the desired size of the dynamic array may be determined. When determining the minimum allowable memory size for a given problem, the dimension of the dynamic array in blank common and the variable NWORDA which supplies this dimension to the various subprograms in the USLOB code must be set to at least

NWORDA = NUMBER * (12 + 7 * NTORSN) + 15*NSUBS + 6

where the parameters NUMBER and NTORSN are identical to the input items of the same names defined in Input Set 1 of Appendix C.

Input Set 2 FORMAT (4E10.0)

DELT, XLOAD, RLOAD, SURCUT

where DELT = size of time step.
XLOAD = longitudinal coordinate, referred to the X-axis of the shell S (Fig. 1), of the point from which the shock wave emanates.
RLOAD = radial polar coordinate, measured from the X-axis of the shell S, of the point from which the shock wave emanates.
SURCUT = surface cutoff time.
SURCUT $\leq$ 0.0 : no surface cutoff effects taken into account.
SURCUT > 0.0 : the incident wave is set to zero at the surface cutoff time specified. This time should correspond to the time at which the surface cutoff wave first reaches the submerged structure.

Note: The user should refer to Appendix C for a discussion of the location of the origin of the circumferential coordinate angle θ.

Input Set 3 for Empirical Pressure-Time History (NCHRG = 1)

```
IF (NCHRG .EQ. 2) GO TO 60
WCHRG, PZMLT, PZEXP, THMLT, THEXP                    FORMAT (5E10.0)
GO TO 65
```

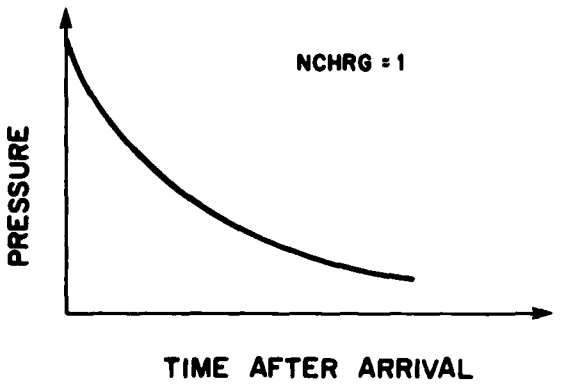
where WCHRG = weight of spherical charge (W, below).
PZMLT = multiplicative constant (k_1, below) for incident pressure.
PZEXP = spatial decay constant (k_2, below) for incident pressure.
THMLT = multiplicative constant (k_3, below) for time constant of exponential decay.
THEXP = spatial decay constant (k_4, below) for time constant of exponential decay.

Note: The above input parameters apply to the empirical pressure-time history described by (Ref. [6])

$$p_I(R,t) = k_1 \ (W^{1/3}/R)^{k_2} \ \exp(-t_a/\theta_o)$$

$$\theta_o(R,t) = k_3 \ W^{1/3} \ (W^{1/3}/R)^{k_4}$$

where p_I is the incident pressure, θ_o is the time constant of exponential decay, R is the distance from the origin of the spherical wave to the point of interest, t represents time, and t_a denotes the time after arrival of the shock wave at the point of interest. The following diagram illustrates the pressure-time history for this option.

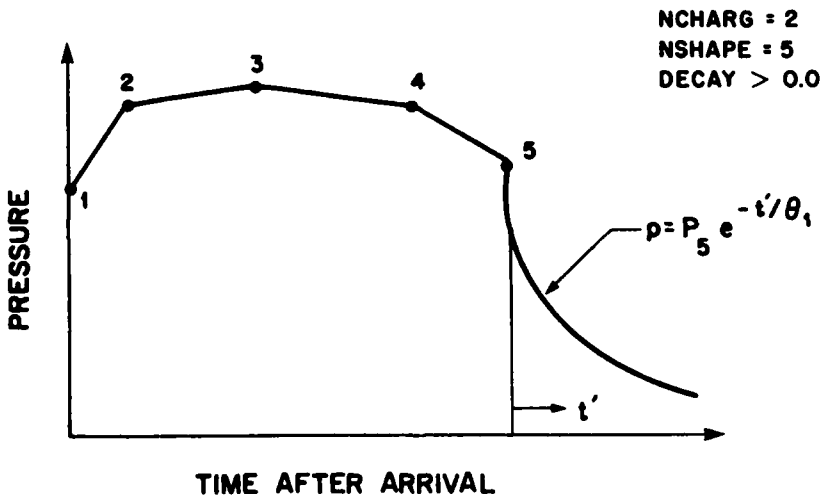


Input Set 3 for Discretized Pressure-Time History (NCHRG = 2)

60 NSHAPE	FORMAT (I5)
(TC(J), PC(J), J = 1, NSHAPE)	FORMAT (8E10.0)
DECAY	FORMAT (E10.0)

where NSHAPE = number of pressure-time points to be furnished for description of incident pressure.
TC = array containing values of time after arrival at which incident pressures are supplied.
PC = array containing values of pressure corresponding to times in array TC.
DECAY = time constant of exponential decay for the decaying exponential tail ending the incident pressure-time history. If a decaying exponential tail is not required, use DECAY $\leq$ 0.0.

Note 1: A typical pressure-time history for this option is shown below.



In the above diagram, the symbols P_5 and θ_1 correspond to the input parameters PC(NSHAPE) and DECAY, respectively.

Note 2: For the current version of USLOB, the input parameter NSHAPE must not exceed 25. This limitation is easily removed by means of the UPDATE batch editing system.

Input Set 4 FORMAT (10I5)

65 (LISTIT(J), J = 1, 20)

where LISTIT(1) = output flag for geometry of shell S.
LISTIT(2) = output flag for natural frequencies and circumferential wave numbers of modes of S.
LISTIT(3) = output flag for normal component of modes of S.
LISTIT(4) = output flag for circumferential wave numbers of surface expansion functions.
LISTIT(5) = output flag for surface expansion functions.
LISTIT(6) = output flag for fluid-shell transformation matrix of Eq. (5).
LISTIT(7) = output flag for virtual mass array.
LISTIT(8) = output flag for inverse of virtual mass array.
LISTIT(9) = output flag for coordinates of nodal points of every substructure.
LISTIT(10) = output flag for shell-substructure transformation matrix of Eq. (9) of every substructure.
LISTIT(11) = output flag for constraint modes of Eq. (10) of every substructure.
LISTIT(12) = output flag for equation numbers or problem unknowns associated with every substructure.
LISTIT(13) = output flag for fixed-base natural frequencies of every substructure.
LISTIT(14) = output flag for fixed-base modes of every substructure.
LISTIT(15) = output flag for portion of modal system mass matrix associated with the shell S after inclusion of effects of shell and concentrated masses.
LISTIT(16) = output flag for expansion coefficients of constraint modes of every substructure.
LISTIT(17) = output flag for portion of modal system mass matrix associated with coupling between the shell S and every substructure.
LISTIT(18) = output flag for interface force coefficients associated with connections between the shell S and every substructure.
LISTIT(19) = output flag for portion of modal system mass matrix associated with the shell S after inclusion of effects of shell, concentrated masses, and every substructure.
LISTIT(20) = output flag for generalized velocities of the shell S and every substructure.

Note: If an output flag is set to 1, output is supplied. If an output flag is set to 0, no output is supplied.

Input Set 5

```
NPTSHL                                                    FORMAT (I5)
(LBOSEG(J), LBOSPT(J), J = 1, NPTSHL)                     FORMAT (12I5)
```

where NPTSHL = number of shell stations at which normal velocity-time histories are to be printed. If no shell responses are desired from USLOB, use NPTSHL $\leq$ 0 and supply no further data for this set of input.

LBOSEG(J) = segment of the shell S in which the J-th velocity station is located.

LBOSPT(J) = mesh point in segment LBOSEG(J) of the shell S at which a normal velocity-time history is to be printed.

Note 1: The normal velocity-time histories at the above shell stations are computed at circumferential coordinate angles θ = 0, 90, 180, 270 deg.

Note 2: Physical points on the shell S are located with reference to a full mathematical model in accordance with the conventions established for the BOSOR4 code.

Input Set 6

```
   DO 70 K = 1, NSUBS
   READ NPTSUB                                            FORMAT (I5)
70 READ (NSTAT(J), NQUAN(J), J = 1, NPTSUB)               FORMAT (12I5)
```

where NPTSUB = number of velocity-time histories which are to be printed for the K-th substructure. If no responses are desired from USLOB for the K-th substructure, use NPTSUB $\leq$ 0 and supply no further data for the K-th substructure.

NSTAT(J) = nodal point of the K-th substructure at which a velocity-time history is to be printed.

NQUAN(J) = identification number of physical degree of freedom at nodal point NSTAT(J) for which a velocity-time history is to be printed.

NQUAN(J) = 1: local x-component of translation.
NQUAN(J) = 2: local y-component of translation.
NQUAN(J) = 3: local z-component of translation.
NQUAN(J) = 4: local x-component of rotation.
NQUAN(J) = 5: local y-component of rotation.
NQUAN(J) = 6: local z-component of rotation.

Note: The substructures are processed sequentially in the order that the substructure files were appended to the shell-fluid file.

5. Sample Control Cards

Shown below is a typical set of control cards or job control language statements for an execution of USLOB under a NOS 1 operating system. A brief description of these control statements is given in the next subsection. Various other arrangements of the job steps are possible.

```
USLOB(T1777) RANLET
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(7777)
ATTACH(OLDPL=USLOB)
UPDATE(F,L=A124)
RETURN(OLDPL)
FTN(I=COMPILE,L=0,OPT=2,PL=30000,A)
RETURN(COMPILE)
FILE(TAPE4,BT=C,RT=S)
ATTACH(TAPE4=PROBONE)
PURGE(PROB105/NA)
DEFINE(TAPE5=PROB105/CT=S,M=R)
PURGE(PROB110/NA)
DEFINE(TAPE10=PROB110/CT=S,M=R)
RFL(145000)
LDSET(PRESET=INDEF,MAP=SB)
LDSET(FILES=TAPE4)
LGO.
GOTO(CATALOG)
EXIT.
CATALOG(TAPE5,R)
CATALOG(TAPE10,R)
```

6. Description of Control Cards

```
USLOB(T1777) RANLET
```
Job card or job statement.
```
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(7777)
```
System dependent accounting information.
```
ATTACH(OLDPL=USLOB)
```
Assign permanent file USLOB to current job with local file name OLDPL. File USLOB contains the UPDATE program library for the USLOB code.
```
UPDATE(F,L=A124)
RETURN(OLDPL)
```
Process UPDATE directives supplied in the job stream (on file INPUT) and prepare a source file suitable for compilation.
```
FTN(I=COMPILE,L=0,OPT=2,PL=30000,A)
RETURN(COMPILE)
```
Produce compiled code or object module for the USLOB code.
```
FILE(TAPE4,BT=C,RT=S)
```
Specify the record type of the indicated local file. This assures that appropriate system routines are loaded for processing.
```
ATTACH(TAPE4=PROBONE)
```
Assign permanent file PROBONE to current job with local file name TAPE4. File PROBONE is the input file for Sample Problem 1.
```
PURGE(PROB105/NA)
DEFINE(TAPE5=PROB105/CT=S,M=R)
PURGE(PROB110/NA)
DEFINE(TAPE10=PROB110/CT=S,M=R)
```
Assign space on permanent file device for the shell-substructure transformation file (TAPE5) and the generalized velocity file (TAPE10) under the permanent file names PROB105 and PROB110, respectively. Information written on TAPE5 and TAPE10 during the execution of USLOB is written directly on the permanent files.
```
RFL(145000)
```
Request field length for execution. This is optional on many systems.
```
LDSET(PRESET=INDEF,MAP=SB)
LDSET(FILES=TAPE4)
LGO.
```
Select load options, load, and execute the compiled version of USLOB. The required set of input data is supplied in the job stream (on file INPUT). The FILES parameter must be supplied on a LDSET control statement for each local file name referenced on a FILE statement.
```
GOTO(CATALOG)
```
Unconditional transfer to first occurrence of control statement CATALOG if the execution terminates normally.
```
EXIT.
```
Terminate execution or provide transfer of control when an execution error occurs.
```
CATALOG(TAPE5,R)
CATALOG(TAPE10,R)
```
List information about each logical record of files TAPE5 and TAPE10 on file OUTPUT.

7. UPDATE Directives and Input Data for Sample Problem 1

The UPDATE directives employed to compute the submerged shock response for Sample Problem 1 are listed below.

```
*IDENT,STORAGE
*DELETE,USLOB.24,USLOB.25
      COMMON A(25000)
      NWORDA = 25000
```

The size of the dynamic array A shown above is somewhat in excess of the size determined by a restricted execution of the USLOB code. In this restricted execution, the dimension of the array A and the variable NWORDA were set to 75 words and the input parameter NTIME was set to 0 (zero). All other input parameters were identical to those listed below. When the dynamic array is dimensioned as above, a field length of almost 145000 words (octal) is required for an execution of the USLOB code.

The set of input data shown below directs the USLOB code to perform the shock response calculation for Sample Problem 1.

```
191     2     2     9    11     1     1    NTIME,NSKIP,NCHRG,NQUAD,NFINE,KOUPLE,NSUBS
2.50E-05  127.8125  855.6125        0.0    DELT,XLOAD,RLOAD,SURCUT
  2                                        NSHAPE
      0.0       10.0        1.0       10.0    (TC(J),PC(J),J=1,NSHAPE)
      0.0                                    DECAY
  0     0     0     0     0     0     0     0    0     0    OUTPUT FLAGS  1-10
  0     0     0     0     0     0     0     0    0     0    OUTPUT FLAGS 11-20
  5                                        NPTSHL
  2     1     2    12     2    22     2    42    2    53    LBOSEG,LBOSPT
  4                                        NPTSUB
  1     3     6     3    15     3    25     3                NSTAT,NQUAN
```

8. UPDATE Directives and Input Data for Sample Problem 2

The UPDATE directives employed to compute the submerged shock response for Sample Problem 2 are listed below.

```
*IDENT,STORAGE
*DELETE,USLOB.24,USLOB.25
      COMMON A(15000)
      NWORDA = 15000
```

The size of the dynamic array A shown above is somewhat in excess of the size determined by a restricted execution of the USLOB code. In this restricted execution, the dimension of the array A and the variable NWORDA were set to 75 words and the input parameter NTIME was set to 0 (zero). All other input parameters were identical to those listed below. When the dynamic array is dimensioned as above, a field length of almost 125000 words (octal) is required for an execution of the USLOB code.

The set of input data shown below directs the USLOB code to perform the shock response calculation for Sample Problem 2.

```
191     2     1     9    11     2     1    NTIME,NSKIP,NCHRG,NQUAD,NFINE,KOUPLE,NSUBS
2.50E-05  127.8125    50.4375        0.0    DELT,XLOAD,RLOAD,SURCUT
    3.0 3.580375+5      1.13 3.357439-5     -0.22 WCHRG,PZMLT,PZEXP,TMMLT,TMEXP
  0     0     0     0     0     0     0     0     0     0    OUTPUT FLAGS  1-10
  0     0     0     0     0     0     0     0     0     0    OUTPUT FLAGS 11-20
  4                                            NPTSML
  1     1     2    19     3     1     3    19                LBOSEG,LBOSPT
  4                                            NPTSUB
  1     3     6     3    15     3    25     3                NSTAT,NQUAN
```

APPENDIX G - THE USE OF PUSLOB

1. General Information

The PUSLOB computer code is included in the ELSHOK suite of computer codes to produce plots of velocity-time histories for any point on the main body S and for any nodal point on any substructure σ. As discussed in Section III, velocity plots for the shell may be provided in the local coordinate system and/or in the global coordinate system. In the local system, the inward normal (n), meridional (s), and circumferential (θ) directions of Fig. 1 are employed. In the global system, the athwartship (-Z), fore-aft (+X), and downward (-Y) directions of Fig. 1 are used. All plotting for a substructure is performed with reference to the coordinate system of the given substructure. In addition to plotted results, coded card-image time histories may also be prepared.

All plotting of velocity-time histories using the PUSLOB code is performed in two stages. The first stage is the execution of PUSLOB to produce a file, called the "plot file", which must be saved on a permanent file device or any other storage unit accessible by means of a TEKTRONIX graphics terminal. The second stage of the operation is the transfer of the plot commands and related data from the saved plot file to a TEKTRONIX graphics terminal (usually via some type of COPY or LIST command) and the subsequent production of the desired time histories on a TEKTRONIX hard copy unit.

2. Rules of Operation

(1) An input file, a generalized velocity file, and a shell-substructure transformation file (for those problems involving at least one substructure) must be available prior to the execution of the PUSLOB code.

(2) The input file must be supplied to PUSLOB as a local file assigned to logical unit 4 and referred to as TAPE4.

(3) The generalized velocity file must be supplied to PUSLOB as the local file TAPE10.

(4) If a shell-substructure transformation file is required for the problem under study, it must be supplied to PUSLOB as the local file TAPE5.

(5) The user must supply control cards or control statements to ensure that the plot file produced by PUSLOB (TAPE99 on logical unit 99) is comprised of system-logical records.

(6) The plot file (TAPE99) must be saved for subsequent use with TEKTRONIX graphics equipment.

(7) The user must decide the disposition of any coded card-image time histories produced by PUSLOB. These time histories are written on a coded file assigned to logical unit 7 and referred to as TAPE7.

3. Description of Input Data

The input data specifications required for an execution of the PUSLOB code are described below. Following Appendix B, these specifications are written in a style similar to FORTRAN.

Input Set 1 FORMAT (5I5)

NTIME, NSKIP, NSUBS, NTEK, NCARD

where NTIME = number of time points to be used in preparing velocity-time histories. The value supplied may not exceed the number used in the execution of the USLOB code which produced the generalized velocity file for the problem under study.

NSKIP = integer parameter for processing generalized velocities. For the current version of PUSLOB, supply the same value of NSKIP used in the execution of the USLOB code which produced the generalized velocity file for the problem under study.

NSUBS = number of substructures. This parameter may be set to zero if no velocity-time histories are desired from PUSLOB for a problem involving substructures.

NTEK = integer parameter indicating whether or not plotted velocity-time histories are to be produced (1=yes; 0=no). If NTEK=0, the plot file (TAPE99) is not produced.

NCARD = integer parameter indicating whether or not coded card-image time histories are to be produced (1=yes; 0=no). If NCARD=0, the file used for the coded velocity-time histories (TAPE7) is not produced.

Note : The current version of PUSLOB employs main memory management, but does not allow a restricted execution to determine the minimum size of the dynamic array needed for a given problem. Thus, the user must not supply a zero or negative value for the input parameter NTIME when executing the PUSLOB code.

Input Set 2 FORMAT (3E10.0)

DELT, XMULT, YMULT

where DELT = size of time step used in execution of USLOB code which produced the generalized velocity file for the problem under study.
XMULT = multiplicative scale factor for time or horizontal axis of every plotted velocity-time history. This parameter allows the units of the time scale to be changed easily.
YMULT = multiplicative scale factor for velocity or vertical axis of every plotted velocity-time history. This parameter allows the units of the velocity scale to be changed easily.

NOTE: The scale factors do not apply to the coded card-image time histories written on local file TAPE7. Velocity-time histories prepared for coded file TAPE7 have units consistent with those employed for the execution of program USLOB.

Input Set 3 FORMAT (3A10)

(LTITLE(J), J = 1,3)

where LTITLE = thirty-character alphameric main title or heading to be placed on every velocity-time history. During the execution of PUSLOB, a ten-character alphameric identification label, supplied by the user for each time history, is appended to the main heading contained in array LTITLE.

Input Set 4 FORMAT (4A10)

(NOTEX(J), J = 1,2), (NOTEY(J), J = 1,2)

where NOTEX = twenty-character alphameric label for time or horizontal axis of every plotted velocity-time history.
NOTEY = twenty-character alphameric label for velocity or vertical axis of every plotted velocity-time history.

INPUT Set 5

```
      DO 90 K = 1,3
      READ NPTSHL                                          FORMAT (I5)
      IF (NPTSHL .LE. 0) GO TO 90
      DO 80 J = 1, NPTSHL
   80 READ NGAGE, LBOSEG, LBOSPT,                          FORMAT (A10,4I5,2E10.0)
           NSPHC, NANG,   ANGDEG
   90 CONTINUE
```

where NPTSHL = number of shell stations at which velocity-time histories are to be prepared using K-th component of modes of S. If no shell responses are required for a given component of velocity, use NPTSHL $\leq$ 0 and supply no further data for that component.
K = 1 : inward normal velocity (w).
K = 2 : meridional velocity (u).
K = 3 : circumferential velocity (v).

NGAGE = ten-character alphameric label, to be appended to main title, for identifying shell station to which a velocity-time history pertains.

LBOSEG = segment of shell S in which the J-th velocity station, of the NPTSHL stations considered, is located.

LBOSPT = mesh point in segment LBOSEG of shell S at which J-th velocity station is located.

NSPHC = dual purpose integer flag indicating (a) whether or not current K-th local component of velocity is to be used for projection into global coordinate system of shell S and (b) whether or not a hard copy (i.e., plot or card-image history) of resulting K-th projected global component of velocity is to be produced. When producing global components of velocity by projection, the hard-copy options indicated by NSPHC for K = 1, 2, and 3 apply to the athwartship , fore-aft, and downward global components of the velocity of S, respectively, at the station of interest.
NSPHC = 0 : do not use current K-th local component of velocity for projection into global coordinate system of S.
NSPHC = 1 : use current K-th local component of velocity for projection into global coordinate system of S, and provide hard copy of resulting K-th global component of velocity.
NSPHC =-1 : use current K-th local component of velocity for projection into global coordinate system of S, but do not provide hard copy of resulting K-th global component of velocity.

NANG = number of angular locations, at current J-th shell station, at which velocity-time histories are to be produced. For the current version of PUSLOB, use NANG = 1 or 2.

ANGDEG = circumferential coordinate angles (in degrees) of above NANG angular locations.

Note 1: When global components of the shell velocity are being produced at a given shell station (NSPHC $\neq$ 0), the labels supplied as NGAGE for K = 1,2, and 3 apply to the athwartship, fore-aft, and downward global components of the velocity of S, respectively.

Note 2: The parameter NSPHC overrides the NTEK and NCARD parameters on a component-by-component basis if and only if global components of the shell velocity are being produced by projection of the local components.

Note 3: When producing global components of shell velocity at a given station, consistent input data must be supplied for that station for each of the local components (K = 1,2,3) of shell velocity. When more than one shell station is being considered, the input parameters for those stations must be arranged in the same sequence for each local component (K = 1,2,3). Input for which NSPHC = 0 may be interspersed with input for which NSPHC = $\pm$ 1, if desired, without upsetting the sequence of data used for projection.

Input Set 6

```
    DO 110 K = 1, NSUBS
    READ NPTSUB                                          FORMAT (I5)
    IF (NPTSUB .LE. 0) GO TO 110
    DO 100 J = 1, NPTSUB
100 READ NGAGE, NSTAT, NDOFPT, NQUAN                    FORMAT (A10,5I5)
110 CONTINUE
```

where NPTSUB = number of stations at which velocity-time histories are to be prepared for the K-th substructure. If no responses are required for the K-th substructure, use NPTSUB $\leq$ 0 and supply no further data for the K-th substructure.

NGAGE = ten-character alphameric label, to be appended to main title, for identifying nodal point of K-th substructure to which a velocity-time history pertains.

NSTAT = nodal point of the K-th substructure at which a velocity-time history is to be printed.

NDOFPT = number of physical degrees of freedom, at current J-th substructure station, for which velocity-time histories are to be produced. For the current version of PUSLOB, use NDOFPT = 1,2, or 3.

NQUAN = identification numbers of above NDOFPT physical degrees of freedom.

NQUAN = 1: local x-component of translation.
NQUAN = 2: local y-component of translation.
NQUAN = 3: local z-component of translation.
NQUAN = 4: local x-component of rotation.
NQUAN = 5: local y-component of rotation.
NQUAN = 6: local z-component of rotation.

Note: The substructures are processed sequentially in the order that the substructure files were appended to the shell-fluid file.

4. Sample Control Cards

A sample set of control cards or job control language statements for an execution of PUSLOB under a NOS 1 operating system is shown below. The next subsection contains a brief description of these control statements. Various other arrangements of the job steps are possible.

```
PUSLOB(T177) RANLET
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(777)
ATTACH(OLDPL=PUSLOB)
UPDATE(F,L=A124)
RETURN(OLDPL)
FTN(I=COMPILE,L=0,OPT=1,A)
RETURN(COMPILE)
FILE(TAPE4,BT=C,RT=S)
FILE(TAPE99,BT=C,RT=S)
ATTACH(TAPE4=PROBONE)
ATTACH(TAPE5=PROB105)
ATTACH(TAPE10=PROB110)
ATTACH(QTEK)
PURGE(PLTFILE/NA)
DEFINE(TAPE99=PLTFILE/CT=S,M=R)
RFL(135000)
LDSET(PRESET=INDEF,MAP=SB,LIB=QTEK)
LDSET(FILES=TAPE4/TAPE99)
LGO.
EXIT.
```

5. Description of Control Cards

```
PUSLOB(T177) RANLET
```
Job card or job statement.
```
USER(RANLETD,PASWORD)
CHARGE(602201R,PROJECT)
SETBSL(777)
```
System dependent accounting information.
```
ATTACH(OLDPL=PUSLOB)
```
Assign permanent file PUSLOB to current job with local file name OLDPL. File PUSLOB contains the UPDATE program library for the PUSLOB code.
```
UPDATE(F,L=A124)
RETURN(OLDPL)
```
Process UPDATE directives supplied in the job stream (on file INPUT) and prepare a source file suitable for compilation.
```
FTN(I=COMPILE,L=0,OPT=1,A)
RETURN(COMPILE)
```
Produce compiled code or object module for the PUSLOB code.
```
FILE(TAPE4,BT=C,RT=S)
FILE(TAPE99,BT=C,RT=S)
```
Specify the record type of each of the indicated local files. This assures that appropriate system routines are loaded for processing.
```
ATTACH(TAPE4=PROBONE)
ATTACH(TAPE5=PROB105)
ATTACH(TAPE10=PROB110)
```
Assign permanent files PROBONE, PROB105, PROB110 to current job with local file names TAPE4, TAPE5, TAPE10, respectively. These files are the input file, the shell-substructure transformation file, and the generalized velocity file, respectively.
```
ATTACH(QTEK)
```
Assign permanent file QTEK to current job with local file name QTEK. File QTEK contains subroutines used for plotting on TEKTRONIX graphics terminals and serves as a library file for satisfying externals required by PUSLOB.
```
PURGE(PLTFILE/NA)
DEFINE(TAPE99=PLTFILE/CT=S,M=R)
```
Assign space on permanent file device for the plot file (TAPE99) under the permanent file name PLTFILE. Information written on TAPE99 during the execution of PUSLOB is written directly on the permanent file. File TAPE99 must be listed on a TEKTRONIX graphics terminal to produce plots of the desired velocity-time histories.
```
RFL(135000)
```
Request field length for execution. This is optional on many systems.
```
LDSET(PRESET=INDEF,MAP=SB,LIB=QTEK)
LDSET(FILES=TAPE4/TAPE99)
LGO.
```
Select load options, load, and execute the compiled version of PUSLOB. The required set of input data is supplied in the job stream (on file INPUT). The FILES parameter must be supplied on a LDSET control statement for each local file name referenced on a FILE control statement.
```
EXIT.
```
Terminate execution.

6. UPDATE Directives and Input Data for Sample Problem 1

The UPDATE directives employed to produce plots of velocity-time histories for Sample Problem 1 are listed below.

```
*IDENT,STORAGE
*DELETE,PUSLOB.19,PUSLOB.20
      COMMON A(25000)
      NWORDA = 25000
```

The size of the dynamic array shown above is somewhat in excess of the size actually required for Sample Problem 1. In general, a good estimate of the size of the dynamic array A for PUSLOB is provided by the size of the dynamic array employed for the execution of the USLOB code for the problem under study. A field length of almost 135000 words (octal) is required for an excution of PUSLOB when the dynamic array is dimensioned as above.

The set of input data shown below directs the PUSLOB code to produce the plot file from which velocity-time plots may be obtained for Sample Problem 1.

```
  191     2     1     1     0          NTIME,NSKIP,NSUBS,NTEK,NCARD
  2.50E-05     1000.0          1.0     DELT,XMULT,YMULT
SAMPLE PROBLEM 1  (PROBONE)           (LTITLE(J),J=1,3)
         TIME (MSEC)     VELOCITY (IN/SEC)     NOTEX,NOTEY
    5          NPTSHL-W
V-ATH 2/01     2     1     1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-ATH 2/12     2    12     1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-ATH 2/22     2    22     1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-ATH 2/42     2    42     1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-ATH 2/53     2    53     1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
    5          NPTSHL-U
V-FWD 2/01     2     1    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-FWD 2/12     2    12    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-FWD 2/22     2    22    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-FWD 2/42     2    42    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-FWD 2/53     2    53    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
    5          NPTSHL-V
V-DWN 2/01     2     1    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-DWN 2/12     2    12    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-DWN 2/22     2    22    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-DWN 2/42     2    42    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-DWN 2/53     2    53    -1     2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
    3          NPTSUB
TIP (06)       6     1     3     NGAGE,NSTAT,NDOFPT,NQUAN
JCN (15)      15     1     3     NGAGE,NSTAT,NDOFPT,NQUAN
MID (25)      25     1     3     NGAGE,NSTAT,NDOFPT,NQUAN
```

7. UPDATE Directives and Input Data for Sample Problem 2

The UPDATE directives employed to produce plots of velocity-time histories for Sample Problem 2 are listed below.

```
*IDENT,STORAGE
*DELETE,PUSLOB.19,PUSLOB.20
      COMMON A(15000)
      NWORDA = 15000
```

The size of the dynamic array shown above is somewhat in excess of the size actually required for Sample Problem 2. In general, a good estimate of the size of the dynamic array A for PUSLOB is provided by the size of the dynamic array employed for the execution of the USLOB code for the problem under study. A field length of almost 115000 words (octal) is required for an execution of PUSLOB when the dynamic array is dimensioned as above.

The set of input data shown below directs the PUSLOB code to produce the plot file from which velocity-time plots may be obtained for Sample Problem 2.

```
  191    2     1    1    0          NTIME,NSKIP,NSUBS,NTEK,NCARD
  2.50E-05     1000.0 0.0833333     DELT,XMULT,YMULT
SAMPLE PROBLEM 2  (PROBTWO)        (LTITLE(J),J=1,3)
         TIME (MSEC)    VELOCITY (FT/SEC)     NOTEX,NOTEY
    2          NPTSHL-W
V-ATH 3/01     3    1    1    2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-ATH 3/19     3   19    1    2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
    2          NPTSHL-U
V-FWD 3/01     3    1   -1    2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-FWD 3/19     3   19   -1    2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
    2          NPTSHL-V
V-DWN 3/01     3    1   -1    2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
V-DWN 3/19     3   19   -1    2        0.0      180.0 NGAGE,SG,PT,NSPHC,NANG,ANGDEG
    3          NPTSUB
TIP (06)       6    1    3     NGAGE,NSTAT,NDOFPT,NQUAN
JCN (15)      15    1    3     NGAGE,NSTAT,NDOFPT,NQUAN
MID (25)      25    1    3     NGAGE,NSTAT,NDOFPT,NQUAN
```

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

Defense Advanced Rsch Proj Agency
ATTN: TIO

Defense Intelligence Agency
ATTN: DB-4C2
ATTN: DB-4C2, C. Wiehle
ATTN: DT-1C
ATTN: DT-2
ATTN: RTS-2A
ATTN: DB-4C3
ATTN: DB-4C, Rsch, Phys Vuln Br
ATTN: DB-4CI

Defense Nuclear Agency
ATTN: SPSS
4 cy ATTN: TITL

Defense Technical Information Center
12 cy ATTN: DD

Field Command
Defense Nuclear Agency, Det 1
Lawrence Livermore Lab
ATTN: FC-1

Field Command
Defense Nuclear Agency
ATTN: FCPR
ATTN: FCT
ATTN: FCTX
ATTN: FCTT, G. Ganong
ATTN: FCTT, W. Summa
ATTN: FCTXE

Interservice Nuclear Weapons School
ATTN: TTV

Joint Strat Tgt Planning Staff
ATTN: NRI-STINFO Library
ATTN: JLA, Threat Applications Div
ATTN: JLTW, Rautenberg
ATTN: JLTW-2
ATTN: DOXT
ATTN: XPFS

Under Sec of Def for Rsch & Engrg
ATTN: Strategic & Space Sys (OS)

DEPARTMENT OF THE ARMY

BMD Advanced Technology Center
ATTN: ICRDABH-X
ATTN: ATC-T

Chief of Engineers
ATTN: DAEN-RDL
ATTN: DAEN-MPE-T

Dep Ch of Staff for Ops & Plans
ATTN: DAMO-NC, Nuc Chem Dir

Dep Ch of Staff for Rsch Dev & Acq
ATTN: DAMA

DEPARTMENT OF THE ARMY (Continued)

Engineer Studies Center
ATTN: DAEN-FES, LTC Hatch

Harry Diamond Laboratories
ATTN: DELHD-NW-P
ATTN: DELHD-TA-L

US Army Concepts Analysis Agency
ATTN: CSSA-ADL

US Army Engineer Ctr & Ft Belvoir
ATTN: ATZA-DTE-ADM

US Army Engineer School
ATTN: ATZA-CDC

US Army Engr Waterways Exper Station
ATTN: R. Whalin
ATTN: WESSE
ATTN: WESSD, J. Jackson
ATTN: J. Strange
ATTN: J. Zelasko
ATTN: F. Brown
ATTN: Library
ATTN: WESSA, W. Flathau
ATTN: WESSS, J. Ballard

US Army Foreign Science & Tech Ctr
ATTN: DRXST-SD

US Army Mat Cmd Proj Mngr for Nuc Munitions
ATTN: DRCPM-Nuc

US Army Material & Mechanics Rsch Ctr
ATTN: DRXMR, J. Mescall
ATTN: Technical Library

US Army Materiel Dev & Readiness Cmd
ATTN: DRCDE-D, L. Flynn
ATTN: DRXAM-TL

US Army Nuclear & Chemical Agency
ATTN: Library

US Army War College
ATTN: Library

USA Military Academy
ATTN: Document Library

USA Missile Command
ATTN: Documents Section
ATTN: DRSMI-RH

DEPARTMENT OF THE NAVY

Naval Civil Engineering Laboratory
ATTN: Code L51, J. Crawford

Naval Coastal Systems Laboratory
ATTN: Code 741

DEPARTMENT OF THE NAVY (Continued)

David Taylor Naval Ship R&D Ctr
ATTN: Code L42-3, Library
ATTN: Code 1700, W. Murray
ATTN: Code 1844
ATTN: Code 177, E. Palmer
ATTN: Code 172
ATTN: Code 1770.1
ATTN: Code 174
ATTN: Code 2740
ATTN: Code 1740.4
ATTN: Code 1740, R. Short
ATTN: Code 173
ATTN: Code 11
ATTN: Code 1740.5
ATTN: Code 1740.6
ATTN: Code 1740.1

Naval Electronic Systems Command
ATTN: PME 117-21

Naval Explosive Ord Disposal Fac
ATTN: Code 504, J. Petrousky

Naval Facilities Engineering Command
ATTN: Code 04B

Naval Material Command
ATTN: MAT 08T-22

Naval Ocean Systems Center
ATTN: Code 013, E. Cooper
ATTN: Code 4471

Naval Postgraduate School
ATTN: Code 69NE
ATTN: Code 1424, Library
ATTN: Code 69SG, Y. Shin

Naval Research Laboratory
ATTN: Code 8403, R. Belsham
ATTN: Code 8440, G. O'Hara
ATTN: Code 6380
ATTN: Code 8100
ATTN: Code 8301
ATTN: Code 8406
ATTN: Code 2627
ATTN: Code 8445
ATTN: Code 8404, H. Pusey

Naval Sea Systems Command
ATTN: SEA-033
ATTN: SEA-323
ATTN: SEA-06J, R. Lane
ATTN: SEA-09G53
ATTN: SEA-55X1
ATTN: SEA-08
ATTN: SEA-0351
ATTN: SEA-9931G

Naval Surface Weapons Center
ATTN: Code F34
ATTN: Code R13
ATTN: Code R10
ATTN: Code U401, M. Kleinerman
ATTN: Code R14
ATTN: Code F31
ATTN: Code R15

DEPARTMENT OF THE NAVY (Continued)

Naval Surface Weapons Center
ATTN: W. Wishard
ATTN: Tech Library & Info Svcs Br

Naval War College
ATTN: Code E-11, Tech Service

Naval Weapons Center
ATTN: Code 343, FKA6A2, Tech Svcs
ATTN: Code 266, C. Austin
ATTN: Code 3263, J. Bowen

Naval Weapons Evaluation Facility
ATTN: G. Binns
ATTN: Code 10
ATTN: Code 210
ATTN: R. Hughes

Naval Weapons Support Center
ATTN: Code 70553, D. Moore

New London Laboratory
ATTN: Code 4494, J. Patel
ATTN: Code 4492, J. Kalinowski

Newport Laboratory
ATTN: Code EM
ATTN: Code 363, P. Paranzino

Ofc of the Deputy Chief of Naval Ops
ATTN: OP 987
ATTN: NOP 982, Tac Air Srf & Ewdev Div
ATTN: NOP 981
ATTN: NOP 654, Strat Eval & Anal Br
ATTN: OP 098T8
ATTN: OP 982E, M. Lenzini
ATTN: OP 957E
ATTN: NOP 953, Tac Readiness Div
ATTN: OP 37
ATTN: OP 225
ATTN: OP 03EG
ATTN: OP 21
ATTN: NOP 951, ASW Div
ATTN: OP 605D5
ATTN: OP 981N1
ATTN: OP 223

Office of Naval Research
ATTN: Code 474, N. Perrone

Strategic Systems Project Office
ATTN: NSP-272
ATTN: NSP-43
ATTN: NSP-273

DEPARTMENT OF THE AIR FORCE

Air Force Institute of Technology
ATTN: Commander
ATTN: Library

Air Force Systems Command
ATTN: DLW

Assistant Chief of Staff
Intelligence
ATTN: IN

DEPARTMENT OF THE AIR FORCE (Continued)

Air Force Weapons Laboratory
ATTN: NTES-G, S. Melzer
ATTN: NTE, M. Plamondon
ATTN: NTES-C, R. Henny
ATTN: SUL
ATTN: NTED

Ballistic Missile Office
ATTN: DEB

Deputy Chief of Staff
Research, Development, & Acq
ATTN: AFRDQI
ATTN: R. Steere

Deputy Chief of Staff
Logistics & Engineering
ATTN: LEEE

Foreign Technology Division
ATTN: NIIS Library
ATTN: TQTD
ATTN: SDBG
ATTN: SDBF, S. Spring

Rome Air Development Center
ATTN: RBES, R. Mair
ATTN: Commander
ATTN: TSLD

Strategic Air Command
ATTN: NRI-STINFO Library

DEPARTMENT OF ENERGY

Albuquerque Operations Office
ATTN: CTID

Office of Military Application
ATTN: OMA/RD&T

Nevada Operations Office
ATTN: Doc Con for Technical Library

OTHER GOVERNMENT AGENCIES

Central Intelligence Agency
ATTN: OSWR/NED
ATTN: OSR/SE/F

Department of the Interior
US Geological Survey
ATTN: D. Roddy

NASA
ATTN: F. Nichols
ATTN: R. Jackson

US Nuclear Regulatory Commission
ATTN: R. Whipp for Div Sec L. Shao

DEPARTMENT OF ENERGY CONTRACTORS

University of California
Lawrence Livermore National Lab
ATTN: S. Erickson

DEPARTMENT OF ENERGY CONTRACTORS (Continued)

Los Alamos National Laboratory
ATTN: R. Whitaker
ATTN: MS530, G. Spillman
ATTN: Reports Library
ATTN: MS634, T. Dowler
ATTN: R. Sanford
ATTN: MS 670, J. Hopkins

Oak Ridge National Laboratory
ATTN: Civil Def Res Proj
ATTN: Central Rsch Library

Sandia National Lab
ATTN: Tech Lib 3141
ATTN: L. Vortman

Sandia National Labs, Livermore
ATTN: Library & Security Classification Div

DEPARTMENT OF DEFENSE CONTRACTORS

Applied Research Associates, Inc
ATTN: D. Piepenburg

Applied Research Associates, Inc
ATTN: B. Frank

BDM Corp
ATTN: T. Neighbors
ATTN: A. Lavagnino
ATTN: Corporate Library

California Institute of Technology
ATTN: T. Ahrens

California Research & Technology, Inc
ATTN: M. Rosenblatt
ATTN: S. Schuster
ATTN: Library
ATTN: K. Kreyenhagen

Columbia University
ATTN: H. Bleich
ATTN: F. Dimaggio

University of Denver
ATTN: Sec Officer for J. Wisotski

Electric Power Research Institute
ATTN: G. Sliter

Electro-Mech Systems, Inc
ATTN: R. Shunk

General Dynamics Corp
ATTN: J. Miller
ATTN: J. Mador
ATTN: M. Pakstys

Kaman AviDyne
ATTN: R. Ruetenik
ATTN: G. Zartarian
ATTN: Library
ATTN: N. Hobbs

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Kaman Sciences Corp
ATTN: Library
ATTN: F. Shelton

Kaman Sciences Corp
ATTN: D. Sachs

Kaman Tempo
ATTN: DASIAC

Karagozian and Case
ATTN: J. Karagozian

Lockheed Missiles & Space Co, Inc
ATTN: Technical Information Center
ATTN: T. Geers
ATTN: B. Almroth

Lockheed Missiles & Space Co, Inc
ATTN: TIC-Library

M & T Company
ATTN: D. McNaight

Management Science Associates
ATTN: K. Kaplan

McDonnell Douglas Corp
ATTN: R. Halprin

NKF Engineering Associates, Inc
ATTN: R. Belsheim

Pacific-Sierra Research Corp
ATTN: H. Brode, Chairman SAGE

Pacifica Technology
ATTN: R. Bjork
ATTN: G. Kent
ATTN: A. Kushner

Physics Applications, Inc
ATTN: C. Vincent

Science Applications, Inc
ATTN: Technical Library

Southwest Research Institute
ATTN: A. Wenzel
ATTN: W. Baker

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Physics International Co
ATTN: L. Behrmann
ATTN: F. Sauer
ATTN: J. Thomsen
ATTN: E. Moore
ATTN: Technical Library

S-CUBED
ATTN: T. Cherry
ATTN: R. Sedgewick
ATTN: D. Grine
ATTN: T. Riney
ATTN: Library
ATTN: K. Pyatt
ATTN: T. McKinley

SRI International
ATTN: G. Abrahamson
ATTN: W. Wilkinson
ATTN: A. Florence

Teledyne Brown Engineering
ATTN: J. Ravenscraft

Tetra Tech, Inc
ATTN: L. Hwang

TRW Electronics & Defense Sector
ATTN: P. Bhuta
ATTN: A. Feldman
ATTN: N. Lipner
ATTN: Technical Information Center
ATTN: D. Jortner
ATTN: B. Sussholtz

TRW Electronics & Defense Sector
ATTN: P. Dai
ATTN: F. Pieper
ATTN: E. Wong
ATTN: G. Hulcher

Weidlinger Assoc, Consulting Engrg
ATTN: J. McCormick
ATTN: M. Baron
4 cy ATTN: R. Vasudevan
4 cy ATTN: D. Ranlet

Weidlinger Associates
ATTN: J. Isenberg

END

FILMED

9-83

DTIC